TU Clausthal
Clausthal University of Technology

Description Logic for Coalitions

İnanç Seylan and Wojciech Jamroga

IfI Technical Report Series **IfI-08-14**

IfI

Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/forschung/technical-reports/
ISSN: 1860-8477

Description Logic for Coalitions

İnanç Seylan and Wojciech Jamroga

Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
seylan@inf.unibz.it
Department of Informatics, Clausthal University of Technology, Germany
wjamroga@in.tu-clausthal.de

Abstract

Coalition Logic (**CL**) is one of the most important formalisms for specification and verification of game-like multi-agent systems. Several extensions of the logic have been studied in the literature. These extensions are usually fusions (independent joins) of **CL** with other modal logics (e.g., temporal, epistemic, dynamic, etc.), and they are generally propositional. In this paper, we propose a game description logic called $\mathbf{CL}_{\mathcal{ALC}}$ which is based on a product of Coalition Logic with the description logic $\mathcal{ALC}$. The new logic allows one to reason about agents' ability to influence first-order structures. We show that the satisfiability problem for $\mathbf{CL}_{\mathcal{ALC}}$ is decidable; we prove this by giving a goal-directed decision procedure for the problem.

Keywords: Strategic logics, description logics, satisfiability, tableaux.

1 Introduction

The knowledge representation and reasoning formalisms for multi-agent systems broadly fall into two categories. In the first category, mental qualities e.g., beliefs, knowledge, desires, intentions, goals, obligations, etc. are ascribed to agents in order to explain the reasons that lead an agent to rational behavior, particularly to communicate with other agents. Such a theory is defined by special type of modal logics called BDI (belief, desire, intention) logics [8, 24, 25]. In general, BDI logics fail to capture the strategic structure of multi-agent systems. Another problem is that it is not possible to verify certain properties of the system itself [35]. The second category of KR formalisms for multi-agent systems – namely strategic logics – deal with these problems.

Coalition logic **CL** [22, 23] formalizes the ability of groups of agents to achieve certain outcomes in strategic games. The central operator of **CL** is $[A]$, with $[A]\varphi$ meaning that group of agents A has a strategy to achieve an outcome state where φ holds. The logic has important applications in the specification and verification of game-like scenarios, social choice mechanisms (e.g., design of voting protocols) etc. The latter group of applications is closely related to the satisfiability problem for **CL**: showing that a **CL** specification φ is satisfiable amounts in most cases to construction of a model (mechanism, protocol) that satisfies φ.

Several extensions of **CL** have been proposed (see [36] and the references therein). The extensions focus on the representation of the following issues:

- Incomplete information: These logics extend **CL** with epistemic operators of knowledge and common knowledge. The resulting systems allow one to model cases in which agents are not fully aware of the choices made by other agents e.g., sealed bid auctions.
- Preferences: These extensions allow to represent agents' preferences and their effect on the outcomes.
- Quantification: Modal operators in standard **CL** consist of a set of agents. While expressing some properties of the system, this results in formulas that are exponentially long in the number of agents in the system. Allowing some form of quantification over coalitions produces more succinct specifications.
- Origins of power: Each agent is given specific propositions to control in the system. An agent or the coalitions of which it is a member derive its/their power from the propositions under its/their control.
- Social laws: Logics in this category allow to constrain the behaviors of the agents in the system according to some social law.
- One can also find logics that combine action formalisms with cooperational ability in the literature [26].

A common feature of the resulting logics is that the underlying language is propositional. In consequence, we can use them only to address simple properties of games and states, but not properties of individual entities that are involved in the game (like people, places, messages, communication channels etc.). For example, the only way to say that agent a can make all the sent messages reach their recipients is to create a proposition that labels all the states where this is the case (e.g., $\mathsf{allSentReceived}$), and then write $[a]\mathsf{allSentReceived}$. Of course, this method of specification is neither elegant nor flexible, and becomes impractical for all but simplest scenarios.

Description Logics (DLs) are logical formalisms for representing the knowledge of an application domain in a structured way [6]. More precisely, DLs allow to describe classes, assign individuals to these classes, and define binary relations on individuals. For instance, we can use DL terms Sent and $\mathsf{Received}$ for the classes of sent and received messages. Then, the DL formula $\mathsf{Sent} \sqsubseteq \mathsf{Received}$ says that every sent message is received too. Note that the dual statement: "*some* sent messages have been received" can be expressed by formula $\neg(\mathsf{Sent} \sqcap \mathsf{Received} = \bot)$.

Description Logics are important because they are *decidable* fragments of first order logic. Our combination of a DL with coalition logic brings first-order perspective to reasoning about coalitional abilities, while keeping it still decidable. Furthermore, DLs have well developed practical decision procedures. Last but not least, they comprise the formal basis of the Semantic Web ontology languages [14]. The Semantic Web is built on the vision of giving explicit meaning to information, making it easier for software agents to automatically process and integrate information available on the Web [7]. Combining agent logics with DLs enables reasoning about how (and by whom) the information can be manipulated, which is potentially interesting for both the agents community and the Semantic Web community.

1.1 Combining CL and DL

In general, one has to answer the following questions when designing such a combination:

1. Where to apply the modal operators? The modal operators could be allowed in front of formulas, concepts, roles, or any combination of these three syntactical structures.
2. How to interpret DL expressions? Object, concept, and role names may be interpreted locally or globally. For example, if the concept name C is interpreted globally, C will represent the same set of individuals in each possible world.
3. How to relate the domains within different worlds? Since possible worlds are now populated by individuals, it is possible to apply some restrictions on the domains (set of individuals) of worlds. For example, in the expanding domain assumption, the domain of an accessible world subsumes the domain of the current world whereas in the constant domain assumption, domains are the same in each world.

Regarding questions 2 and 3, we assume in this paper that the interpretation of terms (objects, concepts, role names) is local, but the domain of interpretation is global (i.e., the same for each possible world). Question 1 requires some more clarification. To this end, Figure 1 shows example formulas from a coalition DL where the modal operators are applied to a formula, concept, and role, respectively. (1) means that a clown can make a person happy. (2) defines a shirt production unit as something from which agent (machine) 1 can produce blue shirts and 2 can produce yellow shirts but they cannot produce both blue and yellow shirts in a single run of the unit (even if they join forces). (3) says that max is a dog such that all the things Bill and

$$[\mathsf{clown}](\mathsf{Person} \sqsubseteq \mathsf{Happy}) \tag{1}$$
$$\mathsf{ShirtProdUnit} = [1]\mathsf{BShirt} \sqcap [2]\mathsf{YShirt} \sqcap [\emptyset]\neg(\mathsf{Bhirt} \sqcap \mathsf{YShirt}) \tag{2}$$
$$\mathsf{max} : \mathsf{Dog} \sqcap \forall[\mathsf{Bill}, \mathsf{Mary}]\mathsf{loves}.\mathsf{Cat} \tag{3}$$

Figure 1: Different application of modal operators to syntactic terms.

Mary can make him love are cats. In our approach, we want to allow for all such combinations of the description and coalitional dimensions.

The logic that we propose in this paper is a product style combination of the description logic $\mathcal{ALC}$ with coalition logic, that is interpreted over constant domain models. The resulting multi-modal logic called $\mathbf{CL}_{\mathcal{ALC}}$ allows for application of modal operators to both formulas and concepts. Again, concept names and role names are interpreted locally. For example, the $\mathbf{CL}_{\mathcal{ALC}}$ formula $[a](\mathsf{Sent} \sqsubseteq \mathsf{Received})$ can be used to express that agent a can make the sent messages be received. Another meaningful $\mathbf{CL}_{\mathcal{ALC}}$ specification $\mathsf{Sent} \sqsubseteq [a]\mathsf{Received}$ says that, for every message that has been *already* sent (prior to a's involvement), a can guarantee its reception. We emphasize that the combination of logics, studied here, is not trivial. This is because $\mathbf{CL}_{\mathcal{ALC}}$ is closely related to the Cartesian product $\mathbf{CL} \times \mathbf{S5}$ (cf. [30, 20] for a more detailed discussion).

1.2 Related Work

Combinations of various modal logics with DLs have been studied extensively, e.g. in [17, 18, 5, 31, 32, 33, 34, 30, 4]. In our previous work, we presented a tableau decision procedure for $\mathcal{ALC}_{\mathcal{BI}}$ which was a fusion style extension of the basic DL $\mathcal{ALC}$ with belief and intention modalities [10]. In $\mathcal{ALC}_{\mathcal{BI}}$, modal operators are only allowed in front of formulas and no restriction is enforced on the domains of worlds. When used in a multi-agent development framework [9], $\mathcal{ALC}_{\mathcal{BI}}$ allowed agents to interpret the meanings of agent communication language messages based on speech acts [11] and then to automatically generate a response based on the interpreted knowledge. Moreover, our agents could process (a subset of) OWL ontologies that were already available to them due to the DL component of $\mathcal{ALC}_{\mathcal{BI}}$. This has motivated us to design similar extensions of DLs with game-theoretic multi-agent logics.

In this paper, we show that the formula satisfiability problem of $\mathbf{CL}_{\mathcal{ALC}}$ is decidable by giving a tableau decision procedure for it. The algorithm presented in this paper is developed incrementally (similarly to the approach in [13]) in the sense that we start with the decision procedure of a simpler logic called $\mathbf{M}_{\mathcal{ALC}}$ [27]. $\mathbf{M}_{\mathcal{ALC}}$ is the logic obtained by combining $\mathcal{ALC}$ with monotonic modal operators that can be applied to formulas and concepts. In [27], we also restricted our attention to constant domain models.

1.3 Structure of the Paper

The paper is organized as follows. First the logic $\mathbf{CL}_{\mathcal{ALC}}$ is introduced. Then we define structures that are equivalent to $\mathbf{CL}_{\mathcal{ALC}}$ models. These structures enable us to reason with $\mathbf{CL}_{\mathcal{ALC}}$ formulas in a more convenient way. Next, we present our tableau based decision procedure for the satisfiability of $\mathbf{CL}_{\mathcal{ALC}}$ formulas and prove its correctness. Finally, we conclude the work.

The construction of the semantic structures begins by defining a *tableau* for our logic. Then, we move on to a *quasitableau*, and finally to a *locally correct tableau* which will be used by our algorithm for satisfiability checking. Note that these structures are all equivalent when it comes to satisfiability. However, in the literature there is no known way of how a tableau decision procedure for a combined modal DL (such as the one presented here) should construct a representation of a (simple) tableau. This is because of termination problems (see [20] for a discussion). In short, our detailed presentation of the structures is meant to provide a smooth transition that solves the termination problem of the algorithm at the semantic level.

2 Coalitional Description Logic

In this section, we introduce our logic $\mathbf{CL}_{\mathcal{ALC}}$. The logic combines the first-order perspective of the basic description logic $\mathcal{ALC}$ with strategic modalities of Coalition Logic. On the syntactical level, $\mathcal{ALC}$ contributes terms for individuals and their classes (i.e., concepts), while $\mathbf{CL}$ adds operators for reasoning about outcome of strategies and dynamics of concepts. On the semantic level, models of $\mathbf{CL}$ (which can be roughly understood as strategic games played successively one after another) are enriched with concept structures that can evolve over time.

2.1 Syntax

Definition 1 *Let $\mathbb{A}\mathrm{gt}$ be a finite non-empty set of agents, and let N_C and N_R be countably infinite sets of concept names and role names, respectively. To every coalition $A \subseteq \mathbb{A}\mathrm{gt}$, the modal operators $[A]$ and $\langle A \rangle$ are associated. $\wedge$, $\vee$, and $\neg$ represent standard logical connectives. Every concept name in N_C, as well as $\top$ (top concept) and $\bot$ (bottom concept) are concepts. Let C and D be concepts, R a role name in N_R, and $A \subseteq \mathbb{A}\mathrm{gt}$. Then $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$, $[A]C$, and $\langle A \rangle C$ are concepts. $C \sqsubseteq D$ and $C = D$ are atomic formulas. If φ and ψ are formulas then so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $[A]\varphi$, and $\langle A \rangle \varphi$.*

That is, atomic formulas are sentences that compare concepts, and modal formulas can refer to coalitional ability to enforce a particular relationship between concepts. E.g., formula $[\{clown, jumbo\}](\mathsf{Person} \sqsubseteq \mathsf{Happy})$ states that the clown and Jumbo can make every person happy, while $\langle\{clown\}\rangle\neg(\mathsf{Person}\sqcap \mathsf{Sad} = \bot)$ says that the clown cannot avoid (on his own) some persons being sad. Concepts are either primitive or built from simpler ones by use of constructors $\neg, \sqcap, \sqcup$, etc. We also add "strategic" concept constructors: $[\{clown\}]\mathsf{Happy}$ reads as "the set of individuals that can be turned happy by the clown", and $\langle\{jumbo\}\rangle\mathsf{Sad}$ as "those that Jumbo cannot prevent from becoming sad".

Other logical connectives, namely $\rightarrow$ and $\leftrightarrow$, can be defined as $\neg\varphi \vee \psi$ and $(\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$, respectively. While writing modal operators, we will omit set braces from coalitions; for instance, we will write $[1, 2]$ instead of $[\{1, 2\}]$. Note that named individuals are not allowed in the DL part of our language for ease of presentation. The interested reader is referred e.g. to [28, 20, 19] to see how named individuals can be dealt with.

2.2 Semantics

The semantics of $\mathbf{CL}_{\mathcal{ALC}}$ joins the first-order interpretation of concepts from $\mathcal{ALC}$ with the possible world semantics of $\mathbf{CL}$ operators. The interpretation of a concept can evolve over time as a result of strategic choices of agents. However, we assume for simplicity that the domain of interpretation does not change from state to state.

Definition 2 *A coalition model for $\mathbf{CL}_{\mathcal{ALC}}$ is a triple of the form $\mathfrak{M} = \langle W, E, \mathcal{I}\rangle$, where W is a non-empty set of states, E is a map associating with each $w \in W$ a playable[1] effectivity function $E_w : 2^{\mathbb{A}\mathrm{gt}} \rightarrow 2^{2^W}$, and $\mathcal{I}$ is a function associating with each $w \in W$ an $\mathcal{ALC}$ interpretation $\mathcal{I}(w) = \left\langle \Delta^{\mathcal{I}(w)}, \cdot^{\mathcal{I}(w)} \right\rangle$. An element V of $E_w(A)$ i.e., a subset of W, is called an outcome. $\Delta^{\mathcal{I}(w)}$ is a non-empty set called the domain of state w, and $\cdot^{\mathcal{I}(w)}$ maps each concept name C to a subset $C^{\mathcal{I}(w)}$ of $\Delta^{\mathcal{I}(w)}$ and each role name R to a binary relation $R^{\mathcal{I}(w)}$ on $\Delta^{\mathcal{I}(w)}$. For any $w, v \in W$, we have $\Delta^{\mathcal{I}(w)} = \Delta^{\mathcal{I}(v)}$ (constant domain assumption).*

Let V be an outcome and A a coalition. The complements of these sets are denoted by $\overline{V}$ and $\overline{A}$ respectively i.e., $\overline{V} = W \setminus V$ and $\overline{A} = \mathbb{A}\mathrm{gt} \setminus A$.

Definition 3 ([22]) *An effectivity function E_w is* playable *iff it satisfies the following conditions:*

(C1) E_w is serial*: $\emptyset \notin E_w(A)$ for all coalitions A.*

*(C2) E_w is W-*complete*: $W \in E_w(A)$. for all coalitions A.*

[1] See below for the definition of playability.

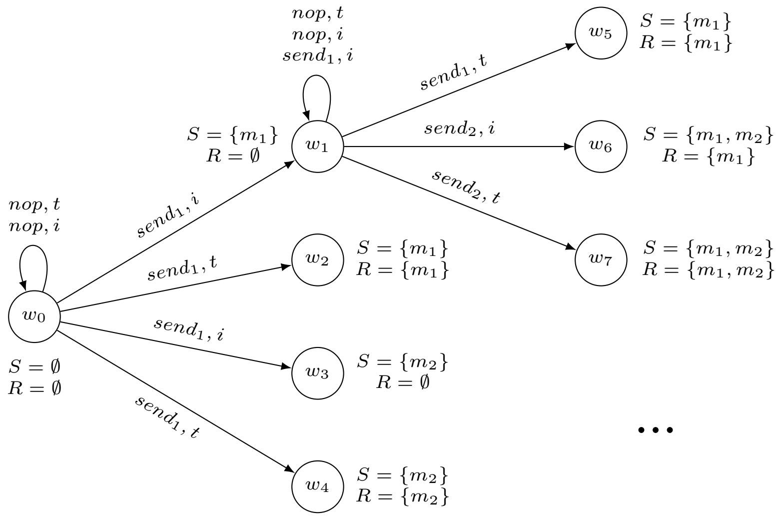

Figure 2: Sending messages through a fitful channel

(C3) E_w *is* $\mathbb{A}\mathrm{gt}$-maximal*: for all* V*, if* $\overline{V} \notin E_w(\emptyset)$ *then* $V \in E_w(\mathbb{A}\mathrm{gt})$*.*

(C4) E_w *is* outcome-monotonic*: for all* $V \subseteq U \subseteq W$ *and for all* A*, if* $V \in E_w(A)$ *then* $U \in E_w(A)$*.*

(C5) E_w *is* superadditive*: for all* V*,* U*,* A_1*, and* A_2 *such that* $A_1 \cap A_2 = \emptyset$*, if* $V \in E_w(A_1)$ *and* $U \in E_w(A_2)$ *then* $V \cap U \in E_w(A_1 \cup A_2)$*.*

Example 1 *Consider a system that consists of a sender* s*, a receiver* r*, and a communication channel* c*. The sender can either do nothing (action* nop*), or send message* m_1 *(action* $send_1$*) or* m_2 *(action* $send_2$*). The channel can either transmit the message to the receiver (action* t*) or ignore it (action* i*). We assume that the receiver receives incoming messages automatically. Thus, he does not execute any relevant actions, and it is enough to include only actions of the sender and the channel in the model. The action structure of the system is depicted in Figure 2 (note: only outgoing transitions from states* w_0 *and* w_1 *are shown). The representation of agents' choices in the graph is somewhat more compact than in Definition 2;*[2] *however, it is easy to read the effectivity functions from the picture by looking at each action (resp. combination of actions) and collecting the states to which the action can lead (and then including also all supersets of the outcomes already present on the list). For example,* $E_{w_0}(c)$ *includes* $\{w_0, w_1, w_3\}$ *(outcome of ignoring),* $\{w_0, w_2, w_4\}$ *(outcome of transmitting), plus all their supersets in* W*.*

[2] In fact, the graph is what Pauly calls a *multi-player game model* [22].

The domain of interpretation contains only messages, i.e., $\Delta^{\mathcal{I}(w)} = \{m_1, m_2\}$ *for all* w. *There are two primitive concepts:* Sent *and* Received *that are used to register the messages that have been sent (resp. received) until the current moment. The interpretation of* Sent *(*$\mathsf{Sent}^{\mathcal{I}(w)}$*) is denoted by* S *in the graph;* $\mathsf{Received}^{\mathcal{I}(w)}$ *is referred to with* R.

It is possible to define playability through a different list of conditions. Such a characterization will help in the coming proofs.

Definition 4 *An effectivity function* E_w *is:*

- Semi-playable *iff it is serial for all* $A \neq \mathbb{A}\mathrm{gt}$*,* W*-complete for all* $A \neq \mathbb{A}\mathrm{gt}$*, outcome-monotonic for all* $A \neq \mathbb{A}\mathrm{gt}$*, and superadditive for all* $A_1 \neq \mathbb{A}\mathrm{gt}, A_2 \neq \mathbb{A}\mathrm{gt}$.
- Regular *iff for all* V *and for all* A*, if* $V \in E_w(A)$ *then* $\overline{V} \notin E_w(\overline{A})$.

Proposition 1 *An effectivity function* E_w *is playable iff it is semi-playable, regular, and* $\mathbb{A}\mathrm{gt}$*-maximal.*

Proof We make use of Pauly's proof [23]. For the if direction, assume that E_w is semi-playable, regular, and $\mathbb{A}\mathrm{gt}$-maximal. We prove each playability condition from Definition 3 one by one.

- (C1): It is enough to show that $\emptyset \notin E_w(\mathbb{A}\mathrm{gt})$ because the other cases for A are covered by (C1) of Definition 4. By (C2) of Definition 4 we have $W \in E_w(\emptyset)$. $\emptyset \notin E_w(\mathbb{A}\mathrm{gt})$ then follows immediately from regularity.
- (C2): It is enough to show that $W \in E_w(\mathbb{A}\mathrm{gt})$ because the other cases for A are covered by (C2) of Definition 4. $W \in E_w(\mathbb{A}\mathrm{gt})$ follows immediately from (C1) of Definition 4 and $\mathbb{A}\mathrm{gt}$-maximality.
- (C3): $\mathbb{A}\mathrm{gt}$-maximality is already a defined condition.
- (C4): It is enough to show that $E_w(\mathbb{A}\mathrm{gt})$ is outcome-monotonic because the other cases for A are covered by (C4) of Definition 4. Suppose that $U \notin E_w(\emptyset)$. Then for each $V_i \subseteq U$, $V_i \notin E_w(\emptyset)$ (due to (C3) from Definition 4). This means $X \in E_w(\mathbb{A}\mathrm{gt})$ where $X = \overline{U}$, and for each $Y_i \supseteq X$ where $Y_i = \overline{V}_i$ we have $Y_i \in E_w(\mathbb{A}\mathrm{gt})$ (due to $\mathbb{A}\mathrm{gt}$-maximality). Hence, $E_w(\mathbb{A}\mathrm{gt})$ is outcome-monotonic.
- (C5): It is enough to show this condition only for A_1 and A_2 with $A_1 \cup A_2 = \mathbb{A}\mathrm{gt}$ because the other cases are covered by (C4) of Definition 4. Assume $V \in E_w(A_1)$ and $U \in E_w(A_2)$ where $A_1 \cap A_2 = \emptyset$ and $A_1 \cup A_2 = \mathbb{A}\mathrm{gt}$. Consider the only two cases:

  1. Suppose $A_1 = \mathbb{A}\mathrm{gt}$, $A_2 = \emptyset$. Assume by contradiction that $V \cap U \notin E_w(\mathbb{A}\mathrm{gt})$. By $\mathbb{A}\mathrm{gt}$-maximality, $\overline{V \cap U} \in E_w(\emptyset)$ and by (C4) of Definition 4, $\overline{V} \cap U \in E_w(\emptyset)$. By monotonicity, $\overline{V} \in E_w(\emptyset)$ and by regularity, $V \notin E_w(\mathbb{A}\mathrm{gt})$, a contradiction. The symmetric case is treated analogously.

2. Suppose $A_1 = \overline{A_2}$. Assume by contradiction that $V \cap U \notin E_w(\mathbb{A}\mathrm{gt})$. By $\mathbb{A}\mathrm{gt}$-maximality, $\overline{V \cap U} \in E_w(\emptyset)$ and by (C4) of Definition 4, $V \cap \overline{U} \in E_w(A_1)$. By monotonicity, $\overline{U} \in E_w(A_1)$ and by regularity, $U \notin E_w(A_2)$, a contradiction.

For the converse, assume that E_w is playable. Since the conditions for playability (Definition 3) subsume the conditions for semi-playability (Definition 4) and E_w is by definition $\mathbb{A}\mathrm{gt}$-maximal, it is enough to show that E_w is regular. Let $V \in E_w(A)$ and assume by contradiction that $\overline{V} \in E_w(\overline{A})$. By superadditivity, $\emptyset \in E_w(\mathbb{A}\mathrm{gt})$ i.e., a contradiction to (C1) of Definition 3.

The interpretation $\mathcal{I}(w)$ defines the semantics of primitive concepts in state w. We extend it to concept descriptions in the standard DL fashion:

$$
\begin{aligned}
\top^{\mathcal{I}(w)} &= \Delta^{\mathcal{I}(w)}, \\
\bot^{\mathcal{I}(w)} &= \emptyset, \\
(\neg C)^{\mathcal{I}(w)} &= \Delta^{\mathcal{I}(w)} \setminus C^{\mathcal{I}(w)}, \\
(C \sqcap D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \cap D^{\mathcal{I}(w)}, \\
(C \sqcup D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \cup D^{\mathcal{I}(w)}, \\
(\forall R.C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \forall \delta' \, (\langle \delta, \delta' \rangle \in R^{\mathcal{I}(w)} \rightarrow \delta' \in C^{\mathcal{I}(w)})\}, \\
(\exists R.C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \exists \delta' \, (\langle \delta, \delta' \rangle \in R^{\mathcal{I}(w)} \wedge \delta' \in C^{\mathcal{I}(w)})\}.
\end{aligned}
$$

Moreover, we add the following definitions:

$$
\begin{aligned}
([A]C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \|C\|_\delta^{\mathfrak{M}} \in E_w(A)\}, \\
(\langle A \rangle C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid W \setminus \|C\|_\delta^{\mathfrak{M}} \notin E_w(A)\},
\end{aligned}
$$

where $\|C\|_\delta^{\mathfrak{M}} = \{w \in W \mid \delta \in C^{\mathcal{I}(w)}\}$ is the set of states that δ belongs to concept C.

Definition 5 *The satisfaction relation $\models$ for $\mathbf{CL}_{\mathcal{ALC}}$ is defined as follows:*

$$
\begin{array}{lll}
\mathfrak{M}, w \models C \sqsubseteq D & \text{iff} & C^{\mathcal{I}(w)} \subseteq D^{\mathcal{I}(w)}, \\
\mathfrak{M}, w \models C = D & \text{iff} & C^{\mathcal{I}(w)} = D^{\mathcal{I}(w)}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \mathfrak{M}, w \not\models \varphi, \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \vee \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models [A]\varphi & \text{iff} & \|\varphi\|^{\mathfrak{M}} \in E_w(A), \\
\mathfrak{M}, w \models \langle A \rangle\varphi & \text{iff} & W \setminus \|\varphi\|^{\mathfrak{M}} \notin E_w(A),
\end{array}
$$

where $\|\varphi\|^{\mathfrak{M}} = \{w \in W \mid \mathfrak{M}, w \models \varphi\}$ is the set of states that satisfy φ in $\mathfrak{M}$.

Example 2 *For the system from Example 1, we have for instance $w_0 \models [c](\mathsf{Sent} \sqsubseteq \mathsf{Received})$: the channel can guarantee that all sent messages will be received. However, the same property does not hold for some other states (e.g., $w_1 \not\models [c](\mathsf{Sent} \sqsubseteq$*

Received)*) because the channel is memoryless and does not buffer undelivered messages. Other important properties are:* $([\emptyset]\mathsf{Sent}) = \mathsf{Sent}$ *(messages that are guaranteed to be labeled as sent in the next step are exactly those that have been sent until now) and* $([s]\mathsf{Sent}) = \top$ *(s is free to send any message); both formulas hold in every state of the system.*

The following pair of formulas demonstrates the distinction between $[A]$ *as a coalitional modality vs. concept constructor.* $w_0 \models ([s,c]\mathsf{Received}) = \top$*: every message can be transmitted successfully if the sender and the channel cooperate; however,* $w_0 \not\models [s,c](\mathsf{Received} = \top)$*: s and c cannot transmit* all *messages at once.*

A formula φ is *satisfiable* if there exist a model $\mathfrak{M} = \langle W, E, \mathcal{I}\rangle$ and a state $w \in W$ such that $\mathfrak{M}, w \models \varphi$. A concept C is *satisfiable* if there exist $\mathfrak{M} = \langle W, E, \mathcal{I}\rangle$ and $w \in W$ such that $C^{\mathcal{I}(w)} \neq \emptyset$. Concept D *subsumes* concept C if $C^{\mathcal{I}(w)} \subseteq D^{\mathcal{I}(w)}$ for all models $\mathfrak{M} = \langle W, E, \mathcal{I}\rangle$ and all $w \in W$. Note that concept subsumption and concept satisfiability can be reduced to formula (un)satisfiability. Concept C is satisfiable iff formula $\neg(C \sqsubseteq \bot)$ is satisfiable and concept D subsumes concept C iff formula $\neg(C \sqsubseteq D)$ is unsatisfiable. The formula $C \sqsubseteq D$ is clearly equivalent to $\neg C \sqcup D = \top$, and $C = D$ to $(\neg C \sqcup D) \sqcap (\neg D \sqcup C) = \top$. In the remainder of this paper, we will assume without loss of generality that every atomic formula is of the form $E = \top$ and we will restrict our attention to satisfiability of formulas.

Example 3 *The satisfiability problem for formula* $[c](\mathsf{Sent} \sqsubseteq \mathsf{Received}) \wedge ([s]\mathsf{Sent}) = \top \wedge ([s,c]\mathsf{Received}) = \top$ *asks about the existence of a model in which agent c can guarantee that all sent messages will be received, agent s is free to send any message, and every message can be transmitted successfully if s and c cooperate.*

We observe that, as models of $\mathbf{CL}_{\mathcal{ALC}}$ can be seen as a class of (possibly evolving) strategic games, the satisfiability problem for $\mathbf{CL}_{\mathcal{ALC}}$ comes very close to that of *mechanism design*, where one seeks a set of rules that guarantees desirable behavior of agents and of the whole system.

3 Tableaux for $\mathrm{CL}_{\mathcal{ALC}}$

In this section, we define structures called tableaux and show their equivalences to $\mathbf{CL}_{\mathcal{ALC}}$ models. We proceed incrementally: first, we get rid of effectivity functions and then we define more useful abstractions of constant domain models. The structure we get at the end which is called a locally correct tableau is almost directly mappable to the data structure that the algorithm uses. Such abstractions of models are commonly used in devising decision procedures [16].

The way we proceed, and the proofs we make along the way, are very similar to the work of Lutz et al. [20, 19] which also establishes a methodology for designing tableau decision procedures for modal DLs with constant domains. Our main deviation point is that Lutz et al. utilize constraint systems (i.e., the data structures directly used in the tableau algorithm) from the beginning whereas we postpone the introduction of constraint systems until later. This is the result of using the tableau abstraction.

To reduce the number of tableau properties, we assume all formulas and concepts to be in *negation normal form* (NNF), i.e., negation signs can appear only in front of atomic formulas and concept names. Every formula (and concept) can be transformed into an equivalent one in NNF by making use of de Morgan's laws, the duality between value restrictions and full existential quantifications, and between modal operators. The NNFs of a formula φ and a concept C are denoted by $\dot{\neg}\varphi$ and $\dot{\neg}C$, respectively.

For a $\mathbf{CL}_{\mathcal{ALC}}$ formula φ, denote by

- $con(\varphi)$ the set of all concepts occurring in φ,
- $rol(\varphi)$ the set of all role names occurring in φ,
- $\mathbb{A}\mathrm{gt}$ the set of all agents occurring in φ,
- $for\varphi)$ the set of all subformulas of φ,
- $con^{\dot{\neg}}(\varphi) = con(\varphi) \cup \{\dot{\neg}C \mid C \in con(\varphi)\}$,
- $for{+}(\varphi) = for\varphi) \cup \{[\emptyset]\vartheta \mid \langle\mathbb{A}\mathrm{gt}\rangle\vartheta \in for\varphi)\}$,
- $con^{+}(\varphi) = con^{\dot{\neg}}(\varphi) \cup \{[\emptyset]C \mid \langle\mathbb{A}\mathrm{gt}\rangle C \in con^{\dot{\neg}}(\varphi)\}$.

3.1 A Tableau for $\mathrm{CL}_{\mathcal{ALC}}$

Definition 6 *If φ is a $\mathbf{CL}_{\mathcal{ALC}}$ formula, a pre-tableau for φ is defined to be a pentuple $\langle\Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}\rangle$ such that*

- *Σ is a non-empty set of states,*
- *$\Lambda : \Sigma \to 2^{for+(\varphi)}$ maps each state to a set of formulas which is a subset of $for{+}(\varphi)$,*
- *$\mathbf{S}$ is a non-empty set of individuals,*
- *$\mathcal{L}$ associates with each state $w \in \Sigma$ a function*

$$\mathcal{L}_w : \mathbf{S} \to 2^{con^{+}(\varphi)}$$

  that maps each individual s in $\mathbf{S}$ to a set of concepts which is a subset of $con^{+}(\varphi)$,
- *$\mathcal{E}$ associates with each state $w \in \Sigma$ a function*

$$\mathcal{E}_w : rol(\varphi) \to 2^{\mathbf{S}\times\mathbf{S}}$$

that maps each role R in $rol(\varphi)$ to a set of pairs of individuals,

- *there is some $w_\varphi \in \Sigma$ such that $\varphi \in \Lambda(w_\varphi)$.*

A pre-tableau T for φ must satisfy some properties so that its equivalence to a model satisfying φ can be shown. These properties make use of the structure of concepts and formulas. The problem is that modal concepts always interact with modal formulas and defining the same property three times (once for a set of modal concepts, once for a set of modal formulas, and once for a set of modal concepts and modal formulas) is unnecessary. Therefore, we will use some notational convenience.

Definition 7 *Let $\langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E} \rangle$ be a pre-tableau for φ. For a state $w \in \Sigma$, the set Φ_w is defined as*

$$\Phi_w = \{\vartheta \mid \vartheta \in \Lambda(w)\} \cup \{s : C \mid C \in \mathcal{L}_w(s) \text{ and } s \in \mathbf{S}\}.$$

α and β are placeholders for elements of a set of the form Φ_w. The expression $[A]\alpha$ is either equal to some $[A]\vartheta$ or $s : [A]C$, and $\langle A \rangle \alpha$ to some $\langle A \rangle \vartheta$ or $s : \langle A \rangle C$. If $[A]\alpha = [A]\vartheta$ or $\langle A \rangle \alpha = \langle A \rangle \vartheta$, then $\alpha = \vartheta$; and if $[A]\alpha = s : [A]C$ or $\langle A \rangle \alpha = s : \langle A \rangle C$, then $\alpha = s : C$.

As a reader with tableau background would notice, the meanings of the symbols α and β in our unifying notation are different than in Smullyan's α and β notation to classify formulas. We are now in a position to define the properties of a tableau for φ.

Definition 8 *Let $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E} \rangle$ be a pre-tableau for φ. T is said to be a tableau for φ if for all $w \in \Sigma$, $s, t \in \mathbf{S}$, $\vartheta, \vartheta_1, \vartheta_2 \in for{+}(\varphi)$, $C, C_1, C_2 \in con^+(\varphi)$, $R \in rol(\varphi)$, $A, A_1, \ldots, A_n \subseteq \mathbb{A}\mathrm{gt}$, it holds that:*

- *(P1) if $C \in \mathcal{L}_w(s)$, then $\neg C \notin \mathcal{L}_w(s)$,*
- *(P2) if $C_1 \sqcap C_2 \in \mathcal{L}_w(s)$, then $C_1 \in \mathcal{L}_w(s)$ and $C_2 \in \mathcal{L}_w(s)$,*
- *(P3) if $C_1 \sqcup C_2 \in \mathcal{L}_w(s)$, then $C_1 \in \mathcal{L}_w(s)$ or $C_2 \in \mathcal{L}_w(s)$,*
- *(P4) if $\forall R.C \in \mathcal{L}_w(s)$ and $\langle s, t \rangle \in \mathcal{E}_w(R)$, then $C \in \mathcal{L}_w(t)$,*
- *(P5) if $\exists R.C \in \mathcal{L}_w(s)$, then there is some $s' \in \mathbf{S}$ such that $\langle s, s' \rangle \in \mathcal{E}_w(R)$ and $C \in \mathcal{L}_w(s')$,*
- *(P6) if $C = \top \in \Lambda(w)$, then $C \in \mathcal{L}_w(s)$,*
- *(P7) if $\neg(C = \top) \in \Lambda(w)$, then there is some $s' \in \mathbf{S}$ such that $\dot\neg C \in \mathcal{L}_w(s')$,*
- *(P8) if $\vartheta \in \Lambda(w)$, then $\neg\vartheta \notin \Lambda(w)$,*
- *(P9) if $\vartheta_1 \wedge \vartheta_2 \in \Lambda(w)$, then $\vartheta_1 \in \Lambda(w)$ and $\vartheta_2 \in \Lambda(w)$,*
- *(P10) if $\vartheta_1 \vee \vartheta_2 \in \Lambda(w)$, then $\vartheta_1 \in \Lambda(w)$ or $\vartheta_2 \in \Lambda(w)$,*
- *(P11) if $\langle \mathbb{A}\mathrm{gt} \rangle \alpha \in \Phi_w$, then $[\emptyset]\alpha \in \Phi_w$,*

(P12) *if* $[A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ *such that* $a \in A_i \cap A_j$ *implies* $i = j$*, then there is* $v \in \Sigma$ *such that* $\alpha_1, \ldots, \alpha_n \in \Phi_v$*,*

(P13) *if* $\langle A \rangle \alpha, [A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ *such that* $a \in A_i \cap A_j$ *implies* $i = j$ *and* $\bigcup_{i=1}^{n} A_i \subseteq A$*, then there is some* $v \in \Sigma$ *such that* $\alpha, \alpha_1, \ldots, \alpha_n \in \Phi_v$*,*

(P14) *if* $\langle A \rangle \alpha \in \Phi_w$*, then there is* $v \in \Sigma$ *such that* $\alpha \in \Phi_v$*.*

Remark 2 *In properties (P1) and (P8), we use* $\neg C$ *and* $\neg\vartheta$ *instead of their negation normal forms because this suffices for the following lemma.*

Proposition 3 *A* $\mathbf{CL}_{\mathcal{ALC}}$ *formula* φ *is satisfiable iff there exists a tableau for* φ*.*

Proof For the *if* direction, let $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E} \rangle$ be a tableau for φ. Define for $\vartheta \in for{+}(\varphi)$,

$$\lfloor \vartheta \rfloor^T = \{ w \in \Sigma \mid \vartheta \in \Lambda(w) \},$$

and for $C \in con^+(\varphi)$ and $s \in \mathbf{S}$,

$$\lfloor C \rfloor_s^T = \{ w \in \Sigma \mid C \in \mathcal{L}_w(s) \}.$$

Note that for every $w \in \Sigma$, $s \in \mathbf{S}$ if $[A]\vartheta \in \Lambda(w)$ then $\lfloor \vartheta \rfloor^T \neq \emptyset$, and if $[A]C \in \mathcal{L}_w(s)$ then $\lfloor C \rfloor_s^T \neq \emptyset$ because of Property (P12) in Definition 8.

As a notational convenience, let $\lfloor \alpha \rfloor^T$ be equal to $\lfloor \vartheta \rfloor^T$ if $\alpha = \vartheta$, and let it be equal to $\lfloor C \rfloor_s^T$ if $\alpha = s : C$. A coalition model $\mathfrak{M} = \langle W, E, \mathcal{I} \rangle$ in which φ is satisfied can be defined as:

1. $W = \Sigma$.
2. $E_w(A)$ is equal to V such that
   (a) *(Case* $A \neq \mathbb{A}\mathrm{gt}$*)* $V = W$, or $\exists [A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$:
      i. $A \supseteq \bigcup_{i=1}^{n} A_i$,
      ii. $\forall a \in \mathbb{A}\mathrm{gt} \ : \ a \in A_i \cap A_j \Rightarrow i = j$,
      iii. $\bigcap_{i=1}^{n} \lfloor \alpha_i \rfloor^T \subseteq V$;
   (b) *(Case* $A = \mathbb{A}\mathrm{gt}$*)* $\overline{V} \notin E_w(\emptyset)$.
3. $\Delta^{\mathcal{I}(w)} = \mathbf{S}$.
4. $D^{\mathcal{I}(w)} = \{ s \mid D \in \mathcal{L}_w(s) \}$ for all concept names D in $con(\varphi)$.
5. $R^{\mathcal{I}(w)} = \mathcal{E}_w(R)$.

Constant domain assumption is validated by the definition of $\Delta^{\mathcal{I}(w)}$ given above. The following lemmas complete this part of the proof:

Lemma 4 *For all* $w \in W$*,* E_w *is playable.*

Proof of Lemma 4 To show that E_w is playable, we make use of Proposition 1. Thus, it suffices to show that E_w is semi-playable, regular and $\mathbb{A}\mathrm{gt}$-maximal.

- Semi-playability: That E_w is semi-playable follows immediately from its construction and the fact that neither $\lfloor \vartheta \rfloor^T = \emptyset$ nor $\lfloor C \rfloor_s^T = \emptyset$ for any $[A]\vartheta \in \Lambda(w)$ or $[A]C \in \mathcal{L}_w(s)$.
- $\mathbb{A}\mathrm{gt}$-maximality: E_w has simply been defined to be $\mathbb{A}\mathrm{gt}$-maximal.
- Regularity: E_w is regular for inputs $\mathbb{A}\mathrm{gt}$ and $\emptyset$, due to its construction. Thus, it only remains to show regularity for a A such that $A \neq \mathbb{A}\mathrm{gt}$ and $A \neq \emptyset$. Assume $V \in E_w(A)$. If $V = W$, then $\emptyset \notin E_w(\overline{A})$ follows from the already proven semi-playability. If $V \neq W$, then there are $[A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ such that the coalitions are pairwise disjoint, $A \supseteq \bigcup_{i=1}^n A_i$, and $\bigcap_{i=1}^n \lfloor \alpha_i \rfloor^T \subseteq V$. Suppose for contradiction that $\overline{V} \in E_w(\overline{A})$. Since $\overline{V} \neq W$, then there are also $[\mathcal{D}_1]\beta_1, \ldots, [\mathcal{D}_m]\beta_m \in \Phi_w$ such that the coalitions are pairwise disjoint, $\overline{A} \supseteq \bigcup_{j=1}^m \mathcal{D}_j$, and $\bigcap_{j=1}^m \lfloor \beta_j \rfloor^T \subseteq \overline{V}$. It is easy to see that each A_i is pairwise disjoint with each $\mathcal{D}_j$. Hence, by Property (P12) in Definition 8, there is a state $v \in W$ such that $\{\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m\} \subseteq \Phi_v$ i.e., $v \in \bigcap_{i=1}^n \lfloor \alpha_i \rfloor^T \cap \bigcap_{j=1}^m \lfloor \beta_j \rfloor^T$. But this is a contradiction so we can conclude that E_w is regular for the input A.

Lemma 5 *For all* $w \in \Sigma$, $E \in con^+(\varphi)$*, and* $s \in \mathbf{S}$*, if* $E \in \mathcal{L}_w(s)$ *then* $s \in E^{\mathcal{I}(w)}$.

Proof of Lemma 5 Let $E \in \mathcal{L}_w(s)$ with $E \in con^+(\varphi)$. The proof is by induction on the structure of E. The case where E is a concept name, $E = \neg C$, $E = (C_1 \sqcap C_2)$, $E = (C_1 \sqcup C_2)$, $E = \exists R.C$, or $E = \forall R.C$ is as presented in [15]. Therefore, only the modal cases will be shown.

1. Let $E = ([A]C)$. We distinguish two cases.
   (a) Assume that $A \neq \mathbb{A}\mathrm{gt}$. By the construction of E_w, $\lfloor C \rfloor_s^T \in E_w(A)$, and by the inductive hypothesis $\lfloor C \rfloor_s^T \subseteq \|C\|_s^{\mathfrak{M}}$. Hence, it follows from outcome-monotonicity of E_w and the semantics of concept expressions that $s \in ([A]C)^{\mathcal{I}(w)}$.
   (b) Assume that $A = \mathbb{A}\mathrm{gt}$. We must show that $\|C\|_s^{\mathfrak{M}} \in E_w(\mathbb{A}\mathrm{gt})$ but suppose for contradiction that $\|C\|_s^{\mathfrak{M}} \notin E_w(\mathbb{A}\mathrm{gt})$. By $\mathbb{A}\mathrm{gt}$-maximality, we have $W \setminus \|C\|_s^{\mathfrak{M}} \in E_w(\emptyset)$. Then there is some $[\emptyset]\alpha_1, \ldots, [\emptyset]\alpha_n \in \Phi_w$ such that
   $$\bigcap_{i=1}^n \lfloor \alpha_i \rfloor^T \subseteq W \setminus \|C\|_s^{\mathfrak{M}} \tag{4}$$
   due to the construction of E_w. From this and our assumption $[\mathbb{A}\mathrm{gt}]C \in \mathcal{L}_w(s)$ it follows by Property (P12) in Definition 8 that there is a

state $v \in W$ such that $\{s : C, \alpha_1, \ldots, \alpha_n\} \subseteq \Phi_v$ i.e., $v \in \lfloor C \rfloor_s^T$ and $v \in \bigcap_{i=1}^n \lfloor \alpha_i \rfloor^T$. By induction, $v \in \|C\|_s^{\mathfrak{M}}$ and from (4), $v \in W \setminus \|C\|_s^{\mathfrak{M}}$. Hence, we found the contradiction.

2. Let $E = (\langle A \rangle C)$. We distinguish two cases, again.

   (a) Assume that $A \neq \mathbb{A}\mathrm{gt}$. In order to show that $s \in (\langle A \rangle C)^{\mathcal{I}(w)}$ i.e., $W \setminus \|C\|_s^{\mathfrak{M}} \notin E_w(A)$, we prove that for all $V \in E_w(A)$ there is a $u \in V$ such that $s \in C^{\mathcal{I}(u)}$. Since $\langle A \rangle C \in \mathcal{L}_w(s)$, there is a $v \in \Sigma$ such that $C \in \mathcal{L}_v(s)$ (due to (P14) of Definition 8). So by the inductive hypothesis, $s \in C^{\mathcal{I}(v)}$. If $V = W$, then V clearly contains v. In this case, just take $u = v$. If $V \neq W$, then there are $[A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ such that the coalitions are pairwise disjoint, $A \supseteq \bigcup_{i=1}^n A_i$, and $\bigcap_{i=1}^n \lfloor \alpha_i \rfloor^T \subseteq V$. But then there must be a state $v' \in \Sigma$ such that $\{s : C, \alpha_1, \ldots, \alpha_n\} \subseteq \Phi_{v'}$ (due to (P13) of Definition 8). By induction, $s \in C^{\mathcal{I}(v')}$. So in this case, take $u = v'$. Since there is no other alternative for V, the case is proved.

   (b) Assume that $A = \mathbb{A}\mathrm{gt}$. Then $[\emptyset]C \in \mathcal{L}_w(s)$ (due to (P11) of Definition 8). By Case 1a, it follows that $s \in ([\emptyset]C)^{\mathcal{I}(w)}$, so by the construction of E_w, we may conclude that $s \in (\langle \mathbb{A}\mathrm{gt} \rangle C)^{\mathcal{I}(w)}$.

Lemma 6 *For every $w \in \Sigma$ and $\psi \in for+(\varphi)$, if $\psi \in \Lambda(w)$ then $\mathfrak{M}, w \models \psi$.*

Proof of Lemma 6 This is also proved by induction.

1. Let ψ be atomic i.e., $\psi = (C = \top)$. Then for each element $s \in \mathbf{S}$, we should have $C \in \mathcal{L}_w(s)$ (due to (P6) of Definition 8). It follows from Claim 2 that $s \in C^{\mathcal{I}(w)}$. Hence $\mathfrak{M}, w \models C = \top$. Next, let $\psi = \neg(C = \top)$. Then there exists an individual $s \in \mathbf{S}$ with $\dot\neg C \in \mathcal{L}_w(s)$ (due to (P7) of Definition 8). It follows from Claim 2 that $s \in (\dot\neg C)^{\mathcal{I}(w)}$. Hence $\mathfrak{M}, w \models \neg(C = \top)$.
2. The proof of the case where ψ is equal to some $\vartheta_1 \wedge \vartheta_2$, $\vartheta_1 \vee \vartheta_2$, $[A]\vartheta$, or $\langle A \rangle \vartheta$ is analogous to its concept counterpart.

For the converse, if $\mathfrak{M} = \langle W, E, \mathcal{I} \rangle$ is a model in which φ is satisfied, then a pre-tableau $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E} \rangle$ for φ can be defined as:

$$
\begin{aligned}
\Sigma &= W \\
\mathbf{S} &= \Delta^{\mathcal{I}(w)} \\
\mathcal{L}_w(s) &= \{C \in con^+(\varphi) \mid s \in C^{\mathcal{I}(w)}\} \\
\mathcal{E}_w(R) &= R^{\mathcal{I}(w)} \\
\Lambda(w) &= \{\psi \in for+(\varphi) \mid \mathfrak{M}, w \models \psi\}
\end{aligned}
$$

It only remains to demonstrate that T is a tableau for φ.

1. T satisfies properties (P1)-(P10) in Definition 8 as a direct consequence of the semantics of their respective concepts and formulas.
2. Let $\langle \mathbb{A}\mathrm{gt} \rangle \alpha \in \Phi_w$. Then $W \setminus \lfloor \alpha \rfloor^T \notin E_w(\mathbb{A}\mathrm{gt})$. From the $\mathbb{A}\mathrm{gt}$-maximality of E_w, we have $\lfloor \alpha \rfloor^T \in E_w(\emptyset)$, and thus $[\emptyset]\alpha \in \Phi_w$. T therefore satisfies Property (P11) in Definition 8.
3. Let $[A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ such that the coalitions are pairwise disjoint. According to the semantics of modal concepts and modal formulas, $\lfloor \alpha_i \rfloor^T \in E_w(A_i)$ for $1 \leq i \leq n$. Since E_w is superadditive, $V = \bigcap_{i=1}^{n} \lfloor \alpha_i \rfloor$ is in $E_w(\mathcal{D})$ where $\mathcal{D} = (\bigcup_{j=1}^{n} A_j)$. By (C1) in Definition 3, $\emptyset \notin E_w(\mathcal{D})$, and thus $V \neq \emptyset$. This means there exists a state $v \in \Sigma$ such that $v \in V$. T therefore satisfies Property (P12) in Definition 8.
4. Let $\langle A \rangle \alpha, [A_1]\alpha_1, \ldots, [A_n]\alpha_n \in \Phi_w$ such that the coalitions are pairwise disjoint and $\bigcup_{i=1}^{n} A_i \subseteq A$. Then $W \setminus \lfloor \alpha \rfloor^T \notin E_w(A)$. Define V and $\mathcal{D}$ as in the previous case so we have $V \in E_w(\mathcal{D})$ and $\mathcal{D} \subseteq A$. For $A' = A \setminus \mathcal{D}$, (C2) of Definition 3 gives us $W \in E_w(A')$ and by superadditivity, $V \in E_w(A)$. Due to outcome-monotonicity, $V \not\subseteq (W \setminus \lfloor \alpha \rfloor^T)$ which means that there exists a state $v \in \Sigma$ such that $v \in V \cap \lfloor \alpha \rfloor^T$. T therefore satisfies Property (P13) in Definition 8.
5. Let $\langle A \rangle \alpha \in \Phi_w$. Then $W \setminus \lfloor \alpha \rfloor^T \notin E_w(A)$. By (C2) of Definition 3, $W \in E_w(A)$. Thus, $\lfloor \alpha \rfloor^T \neq \emptyset$ which means that there exists a state $v \in W$ such that $v \in \lfloor \alpha \rfloor^T$. T therefore satisfies Property (P14) in Definition 8.

3.2 A Quasitableau for $\mathrm{CL}_{\mathcal{ALC}}$

Representing individuals explicitly in a tableau algorithm for a modal DL is generally problematic. To the best of our knowledge, there is no such algorithm for a constant domain modal extension of $\mathcal{ALC}$ that is similar to the logic considered in this paper. For these reasons, we will use an abstraction of a tableau called quasitableau.

Definition 9 *If φ is a $\mathbf{CL}_{\mathcal{ALC}}$ formula, a pre-quasitableau for φ is defined to be a tuple $\langle \Sigma, \Lambda, \mathbf{S}, \mathbf{R}, \mathcal{L}, \mathcal{E} \rangle$ such that:*

- *Σ is a non-empty set of states,*
- *$\Lambda : \Sigma \rightarrow 2^{for+(\varphi)}$ maps each state to a set of formulas which is a subset of $for+(\varphi)$,*
- *$\mathbf{S}$ is a map associating with each $w \in \Sigma$ a non-empty set of concept types,*
- *$\mathbf{R}$ is a non-empty set of runs and a run r in $\mathbf{R}$ is a function associating with every $w \in \Sigma$ a concept type $r(w)$ in $\mathbf{S}(w)$,*
- *$\mathcal{L}$ associates with each state $w \in W$ a function*

$$\mathcal{L}_w : \mathbf{S}(w) \rightarrow 2^{con^+(\varphi)}$$

that maps each concept type s in $\mathbf{S}(w)$ to a set of concepts which is a subset of $con^+(\varphi)$.

- *$\mathcal{E}$ associates with each state $w \in W$ a function*

$$\mathcal{E}_w : rol(\varphi) \rightarrow 2^{\mathbf{S}(w) \times \mathbf{S}(w)}$$

that maps each role R in $rol(\varphi)$ to a set of pairs of concept types from $\mathbf{S}(w)$.

- *there is some $w_\varphi \in \Sigma$ such that $\varphi \in \Lambda(w_\varphi)$.*

Concept types can be thought of as templates for individuals, and runs as template instantiation mechanisms. The set $\mathbf{R}$ corresponds to the set $\mathbf{S}$ of individuals in a tableau. A run $r \in \mathbf{R}$ keeps track of the concept types that represent an individual in states belonging to Σ. Since $r(w)$ is defined for each $w \in \Sigma$, the individual corresponding to r is represented at each state by a concept type. This satisfies the constant domain assumption.

It will again be convenient to use a unifying notation for modal expressions. However, the structural difference between a tableau and a quasitableau requires the redefinition and also addition of some notions.

Definition 10 *Let $\langle \Sigma, \Lambda, \mathbf{S}, \mathbf{R}, \mathcal{L}, \mathcal{E} \rangle$ be a pre-quasitableau for φ. For a state $w \in \Sigma$, the set Φ_w is defined as*

$$\Phi_w = \{\vartheta \mid \vartheta \in \Lambda(w)\} \cup \{s : C \mid C \in \mathcal{L}_w(s) \textit{ and } s \in \mathbf{S}(w)\}.$$

α and β are placeholders for elements of a set of the form Φ_w. The expression $[A]\alpha$ is either equal to some $[A]\vartheta$ or $s : [A]C$, and $\langle A \rangle \alpha$ to some $\langle A \rangle \vartheta$ or $s : \langle A \rangle C$. Let $\Psi \subseteq \Phi_w$ be a set consisting only of expressions of the form $[A]\alpha$ and/or $\langle A \rangle \alpha$. Then the set $\mathsf{qns}(\Psi)$ is equal to $v \in \Sigma$ such that Φ_v is a superset of the union of the following sets:

1. *$\{\vartheta \mid [A]\vartheta$ (or $\langle A \rangle \vartheta) \in \Psi\}$*
2. *$\{r(v) : C \mid r(w) : [A]C$ (or $r(w) : \langle A \rangle C) \in \Psi$ and $r \in \mathbf{R}\}$*

We are now in a position to define the properties of a quasitableau for φ.

Definition 11 *Let $Q = \langle \Sigma, \Lambda, \mathbf{S}, \mathbf{R}, \mathcal{L}, \mathcal{E} \rangle$ be a pre-quasitableau for φ. Q is said to be a quasitableau for φ if for all $w \in \Sigma$, $s, t \in \mathbf{S}(w)$, $\vartheta, \vartheta_1, \vartheta_2 \in for{+}(\varphi)$, $C, C_1, C_2 \in con^+(\varphi)$, $R \in rol(\varphi)$, and $A, A_1, \ldots, A_n \subseteq \mathbb{A}\mathrm{gt}$, it holds that:*

(P0) there exists a run r in $\mathbf{R}$ such that $r(w) = s$,

(P1) if $C \in \mathcal{L}_w(s)$, then $\neg C \notin \mathcal{L}_w(s)$,

(P2) if $C_1 \sqcap C_2 \in \mathcal{L}_w(s)$, then $C_1 \in \mathcal{L}_w(s)$ and $C_2 \in \mathcal{L}_w(s)$,

(P3) *if* $C_1 \sqcup C_2 \in \mathcal{L}_w(s)$*, then* $C_1 \in \mathcal{L}_w(s)$ *or* $C_2 \in \mathcal{L}_w(s)$*,*

(P4) *if* $\forall R.C \in \mathcal{L}_w(s)$ *and* $\langle s, t\rangle \in \mathcal{E}_w(R)$*, then* $C \in \mathcal{L}_w(t)$*,*

(P5) *if* $\exists R.C \in \mathcal{L}_w(s)$*, then there is some* $s' \in \mathbf{S}(w)$ *such that* $\langle s, s'\rangle \in \mathcal{E}_w(R)$ *and* $C \in \mathcal{L}_w(s')$*,*

(P6) *if* $C = \top \in \Lambda(w)$*, then* $C \in \mathcal{L}_w(s)$*,*

(P7) *if* $\neg(C = \top) \in \Lambda(w)$*, then there is some* $s' \in \mathbf{S}(w)$ *such that* $\dot\neg C \in \mathcal{L}_w(s')$*,*

(P8) *if* $\vartheta \in \Lambda(w)$*, then* $\neg\vartheta \notin \Lambda(w)$*,*

(P9) *if* $\vartheta_1 \wedge \vartheta_2 \in \Lambda(w)$*, then* $\vartheta_1 \in \Lambda(w)$ *and* $\vartheta_2 \in \Lambda(w)$*,*

(P10) *if* $\vartheta_1 \vee \vartheta_2 \in \Lambda(w)$*, then* $\vartheta_1 \in \Lambda(w)$ *or* $\vartheta_2 \in \Lambda(w)$*,*

(P11) *if* $\langle \mathbb{A}\mathrm{gt}\rangle\alpha \in \Phi_w$*, then* $[\emptyset]\alpha \in \Phi_w$*,*

(P12) *if* $\Psi = \{[A_1]\alpha_1, \dots, [A_n]\alpha_n\} \subseteq \Phi_w$ *such that* $a \in A_i \cap A_j$ *implies* $i = j$*, then* $\mathsf{qns}(\Psi) \neq \emptyset$*,*

(P13) *if* $\Psi = \{\langle A\rangle\alpha, [A_1]\alpha_1, \dots, [A_n]\alpha_n\} \subseteq \Phi_w$ *such that* $a \in A_i \cap A_j$ *implies* $i = j$ *and* $\bigcup_{i=1}^{n} A_i \subseteq A$*, then* $\mathsf{qns}(\Psi) \neq \emptyset$*,*

(P14) *if* $\Psi = \{\langle A\rangle\alpha\} \subseteq \Phi_w$*, then* $\mathsf{qns}(\Psi) \neq \emptyset$*.*

(P0) says that each concept type is in the range of some run. Properties (P1)-(P11) are as given in Definition 8 with the only difference that in the context of a quasitableau we have concept types instead of individuals. Properties (P12)-(P14) enforce satisfiability preserving restrictions on the runs in $\mathbf{R}$.

Proposition 7 *Let* φ *be a* $\mathbf{CL}_{\mathcal{ALC}}$ *formula. There exists a quasitableau for* φ *iff there exists a tableau for* φ*.*

Proof For the *if* direction, we proceed as in the technical report version of [19]. Let $T = \langle \Sigma_T, \Lambda_T, \mathbf{S}_T, \mathcal{L}^T, \mathcal{E}^T\rangle$ be a tableau for φ. Then $\varphi \in \Lambda_T(w_\varphi)$ for some $w_\varphi \in \Sigma_T$. Fix $w \in \Sigma_T$. Next define equivalence relations $\sim_w$ on $\mathbf{S}_T$ by putting $s \sim_w s'$ iff $\mathcal{L}_w(s) = \mathcal{L}_w(s')$. Consider the equivalence classes modulo $\sim_w$, abbreviated by $[s]_w$. Obviously, $\{[s]_w \mid s \in \mathbf{S}_T\}$ is finite. Choose for each equivalence class $[s]_w$ a concept type $t_{[s]_w}$. Define mappings γ_w which map concept types $t_{[s]_w}$ to sets of domain objects $s \in \mathbf{S}_T$ in the obvious way i.e., $\gamma_w(t_{[s]_w}) = [s]_w$. A quasitableau $Q = \langle \Sigma_Q, \Lambda_Q, \mathbf{S}_Q, \mathbf{R}, \mathcal{L}^Q, \mathcal{E}^Q\rangle$ can be defined from T with

1. $\Sigma_Q = \Sigma_T$
2. $\Lambda_Q(w) = \Lambda_T(w)$
3. $\mathbf{S}_Q(w) = \{t_{[s]_w} \mid s \in \mathbf{S}_T\}$
4. $\mathbf{R} = \{r_s \mid s \in \mathbf{S}_T$ and $\forall w \in \Sigma_Q, r_s(w) = t_{[s]_w}\}$

5. $\mathcal{L}_w^Q(t_{[s]_w}) = \{C \mid s \in \mathbf{S}_T \text{ and } C \in \mathcal{L}_w^T(s)\}$
6. $\mathcal{E}_w^Q(R) = \{\langle t, t' \rangle \mid \exists \ s \in \gamma_w(t) \text{ and } s' \in \gamma_w(t') \text{ with } \langle s, s' \rangle \in \mathcal{E}_w^T(R)\}$.

It is easy to see that Q satisfies all properties in Definition 11. That $\varphi \in \Lambda_Q(w_\varphi)$ follows from the construction of Λ_Q.

For the converse, if $Q = \langle \Sigma_Q, \Lambda_Q, \mathbf{S}_Q, \mathbf{R}, \mathcal{L}^Q, \mathcal{E}^Q \rangle$ is a quasitableau for φ, then a tableau $T = \langle \Sigma_T, \Lambda_T, \mathbf{S}_T, \mathcal{L}^T, \mathcal{E}^T \rangle$ for φ can be defined as

$$\begin{aligned}
\Sigma_T &= \Sigma_Q \\
\Lambda_T(w) &= \Lambda_Q(w) \\
\mathbf{S}_T &= \{r \mid r \in \mathbf{R}\} \\
\mathcal{L}_w^T(r) &= \{C \mid r \in \mathbf{S}_T \text{ and } C \in \mathcal{L}_w^Q(r(w))\} \\
\mathcal{E}_w^T(R) &= \{\langle r, r' \rangle \mid \langle r(w), r'(w) \rangle \in \mathcal{E}_w^Q(R)\}
\end{aligned}$$

We claim that T is a tableau for φ thus, T must satisfy all properties in Definition 8. Its proof is left as an exercise for the reader. That $\varphi \in \Lambda_T(w_\varphi)$ follows from the construction of Λ_T.

3.3 A Locally Correct Tableau for $\mathrm{CL}_{\mathcal{ALC}}$

It turns out that a more compact representation of a quasitableau is possible by relaxing the definition of a run. We first define this structure called a locally correct tableau. Then we show how it can be turned into a quasitableau (and vice versa).

Definition 12 *If φ is a* $\mathbf{CL}_{\mathcal{ALC}}$ *formula, a locally correct pre-tableau for φ is defined to be a hextuple $\langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ such that*

- *Σ is a non-empty set of states,*
- *$\Lambda : \Sigma \rightarrow 2^{for+(\varphi)}$ maps each state to a set of formulas which is a subset of $for+(\varphi)$,*
- *$\mathbf{S}$ is a map associating with each $w \in \Sigma$ a non-empty set of concept types,*
- *$\mathbf{O}$ is a non-empty set of overruns (short for overloaded runs) and an overrun $o \in \mathbf{O}$ is a function associating with every $w \in \Sigma$ a non-empty set of concept types $o(w)$ which is a subset of $\mathbf{S}(w)$,*
- *$\mathcal{L}$ associates with each state $w \in W$ a function*

$$\mathcal{L}_w : \mathbf{S}(w) \rightarrow 2^{con^+(\varphi)}$$

that maps each concept type s in $\mathbf{S}(w)$ to a set of concepts which is a subset of $con^+(\varphi)$.

- *$\mathcal{E}$ associates with each state $w \in W$ a function*

$$\mathcal{E}_w : rol(\varphi) \rightarrow 2^{\mathbf{S}(w) \times \mathbf{S}(w)}$$

that maps each role R in $rol(\varphi)$ to a set of pairs of concept types from $\mathbf{S}(w)$.

- *there is some $w_\varphi \in \Sigma$ such that $\varphi \in \Lambda(w_\varphi)$.*

For a state $w \in \Sigma$ and an overrun $o \in \mathbf{O}$, $|o(w)|$ is called the overloading factor of o in w.

Σ, Λ, $\mathbf{S}$, $\mathcal{L}$, and $\mathcal{E}$ are as defined in Definition 9. An overrun, in contrast to a run, can associate with a state more than one concept type; thus, enabling concept types to be reused. It is this generalization that makes a locally correct tableau a more compact representation of a model than a quasitableau.

Definition 13 *Let $\langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ be a locally correct pre-tableau for φ. For a state $w \in \Sigma$, the set Φ_w is defined as*

$$\Phi_w = \{\vartheta \mid \vartheta \in \Lambda(w)\} \cup \{s : C \mid C \in \mathcal{L}_w(s) \text{ and } s \in \mathbf{S}(w)\}.$$

α and β are placeholders for elements of a set of the form Φ_w. The expression $[A]\alpha$ is either equal to some $[A]\vartheta$ or $s : [A]C$, and $\langle A \rangle \alpha$ to some $\langle A \rangle \vartheta$ or $s : \langle A \rangle C$. Let $\Psi \subseteq \Phi_w$ be a set consisting only of expressions of the form $[A]\alpha$ and/or $\langle A \rangle \alpha$. Then the set $\mathsf{Ins}(\Psi)$ is equal to $v \in \Sigma$ such that for each $[A]\vartheta$ (or $\langle A \rangle \vartheta$) $\in \Psi$, $\vartheta \in \Phi_v$; and for each s with $s : [A]C$ (or $s : \langle A \rangle C$) $\in \Psi$, there exists a concept type $t \in \mathbf{S}(v)$ with $\{C \mid s : [A]C$ (or $s : \langle A \rangle C) \in \Psi\} \subseteq \mathcal{L}_v(t)$, $s \in o(w)$, and $t \in o(v)$.

Let o be an overrun with $o(w) = \{s, t\}$ and $w \in \Sigma$, i.e., the individual corresponding to o is represented by s and t in w. This doesn't mean that s and t are the same concept types. Properties in the following definition make use of this feature.

Definition 14 *Let $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ be a locally correct pre-tableau for φ. T is said to be a locally correct tableau for φ if for all $w \in \Sigma$, $s, t \in \mathbf{S}(w)$, $\vartheta, \vartheta_1, \vartheta_2 \in for{+}(\varphi)$, $C, C_1, C_2 \in con^+(\varphi)$, $R \in rol(\varphi)$, and $A, A_1, \ldots, A_n \subseteq \mathbb{A}\mathrm{gt}$, it holds that:*

(P0) there exists an overrun o in $\mathbf{O}$ such that $s \in o(w)$,

(P1) if $C \in \mathcal{L}_w(s)$, then $\neg C \notin \mathcal{L}_w(s)$,

(P2) if $C_1 \sqcap C_2 \in \mathcal{L}_w(s)$, then $C_1 \in \mathcal{L}_w(s)$ and $C_2 \in \mathcal{L}_w(s)$,

(P3) if $C_1 \sqcup C_2 \in \mathcal{L}_w(s)$, then $C_1 \in \mathcal{L}_w(s)$ or $C_2 \in \mathcal{L}_w(s)$,

(P4) if $\forall R.C \in \mathcal{L}_w(s)$ and $\langle s, t \rangle \in \mathcal{E}_w(R)$, then $C \in \mathcal{L}_w(t)$,

(P5) if $\exists R.C \in \mathcal{L}_w(s)$, then there is some $s' \in \mathbf{S}(w)$ such that $\langle s, s' \rangle \in \mathcal{E}_w(R)$ and $C \in \mathcal{L}_w(s')$,

(P6) if $C = \top \in \Lambda(w)$, then $C \in \mathcal{L}_w(s)$,

(P7) if $\neg(C = \top) \in \Lambda(w)$, then there is some $s' \in \mathbf{S}(w)$ such that $\dot{\neg}C \in \mathcal{L}_w(s')$,

(P8) if $\vartheta \in \Lambda(w)$, then $\neg\vartheta \notin \Lambda(w)$,

(P9) if $\vartheta_1 \wedge \vartheta_2 \in \Lambda(w)$, then $\vartheta_1 \in \Lambda(w)$ and $\vartheta_2 \in \Lambda(w)$,

(P10) if $\vartheta_1 \vee \vartheta_2 \in \Lambda(w)$, then $\vartheta_1 \in \Lambda(w)$ or $\vartheta_2 \in \Lambda(w)$,

(P11) if $\langle \mathbb{A}\mathrm{gt} \rangle \alpha \in \Phi_w$, then $[\emptyset]\alpha \in \Phi_w$,

(P12) if $\Psi = \{[A_1]\alpha_1, \ldots, [A_n]\alpha_n\} \subseteq \Phi_w$ such that $a \in A_i \cap A_j$ implies $i = j$, then $\mathsf{Ins}(\Psi) \neq \emptyset$,

(P13) if $\Psi = \{\langle A \rangle \alpha, [A_1]\alpha_1, \ldots, [A_n]\alpha_n\} \subseteq \Phi_w$ such that $a \in A_i \cap A_j$ implies $i = j$ and $\bigcup_{i=1}^{n} A_i \subseteq A$, then $\mathsf{Ins}(\Psi) \neq \emptyset$,

(P14) if $\Psi = \{\langle A \rangle \alpha\} \subseteq \Phi_w$, then $\mathsf{Ins}(\Psi) \neq \emptyset$.

(P1)-(P11) are as given in Definition 11 and (P0), (P12)-(P14) are analogous to their counterparts in Definition 11. It is therefore not hard to acknowledge that a quasitableau for φ is also a locally correct tableau for φ because each run in the quasitableau can be seen as an overrun with the overloading factor of one. However, the converse does not hold because there exist cases in which we can't (immediately) define $\mathbf{R}$.

Example 4 *Consider the locally correct tableau $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ for φ with $\Sigma = \{w, v\}$ and $\mathcal{L}_w(s) = \{[1]C, [2]D, [2,3]E\}$, $\mathcal{L}_v(s) = \{C, D\}$, $\mathcal{L}_v(t) = \{C, E\}$ (it does not matter how φ actually looks like). T is obviously a locally correct tableau, but it cannot be a quasitableau for φ: there exists no run r with $r(w) = s$, because whatever choice $r(v) = s$ or $r(v) = t$ we make, (P12) in Definition 11 does not hold. However, it is possible to modify T and convert it into a quasitableau by duplicating the state v with all the necessary mappings.*

The following proposition generalizes the observation we made in the previous example.

Proposition 8 *Let φ be a $\mathbf{CL}_{\mathcal{ALC}}$ formula. There exists a locally correct tableau for φ iff there is a quasitableau for φ.*

Proof The *if* direction is trivial. Let us prove the converse. As in the example above, we construct a quasitableau Q for φ by duplicating states that have overruns with overloading factor greater than one in the given locally correct tableau for φ. The algorithm works as follows.

Let $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ be a locally correct tableau for φ. First, take a "copy" $Q = \langle \Sigma', \Lambda', \mathbf{S}', \mathcal{L}', \mathcal{E}' \rangle$ of T (with $\mathbf{O}$ removed). Then, for each $w \in \Sigma$

and each $o \in \mathbf{O}$, if $|o(w)| = n$ and $n > 1$, then create $n - 1$ "copies" of w i.e., $\{w^{(j)} | w \in \Sigma$ and $1 \leq j \leq n - 1\}$ and add them to Σ'. Set

- $\Lambda'(w^{(j)}) = \Lambda'(w)$,
- $\mathbf{S}'(w^{(j)}) = \mathbf{S}'(w)$,
- $\mathcal{L}'_{w^{(j)}}(s) = \{C | s \in \mathbf{S}'(w^{(j)})$ and $C \in \mathcal{L}'_w(s)\}$,
- $\mathcal{E}'_{w^{(j)}}(R) = \{\langle s, s' \rangle | s, s' \in \mathbf{S}'(w^{(j)})$ and $\langle s, s' \rangle \in \mathcal{E}'_w(R)\}$.

Using the fact that T is a locally correct tableau for φ, it is straightforward to show that Q is also a locally correct tableau for φ by inductively constructing an overrun in Q. Moreover, for each $w \in \Sigma'$ and each overrun o in Q, the overloading factor of o in w is equal to one. Thus, Q is a quasitableau for φ.

The following is an immediate consequence of Propositions 3, 7, and 8.

Theorem 9 *A* $\mathbf{CL}_{\mathcal{ALC}}$ *formula* φ *is satisfiable iff there exists a locally correct tableau for* φ*.*

4 Tableau Algorithm for $\mathrm{CL}_{\mathcal{ALC}}$

From Theorem 9, an algorithm which constructs a (finite) representation of a locally correct tableau for a $\mathbf{CL}_{\mathcal{ALC}}$ formula can be used as a decision procedure for the satisfiability of $\mathbf{CL}_{\mathcal{ALC}}$ formulas. In this section, such an algorithm is described, and we prove its termination, soundness, and completeness.

4.1 Definition of the Algorithm

Let N_V be a set of countably infinite variable names, and $<$ be the well-order relation on N_V, and let φ be a $\mathbf{CL}_{\mathcal{ALC}}$ formula. A *constraint* for φ is (i) a formula in $for{+}(\varphi)$, (ii) an atom of the form $x : C$ where $x \in N_V$ and C is a concept in $con^+(\varphi)$, or (iii) an atom of the form $(x, y) : R$ where $x, y \in N_V$ and R is a role in $rol(\varphi)$. A *constraint system* S for φ is a finite, non-empty set of constraints for φ. A *completion set* $\mathbf{T}$ for φ is a set of constraint systems for φ.

Syntactically, our constraints are not very different from those used in standard DL tableau algorithms. The major difference is in their semantics because variables in standard DL tableau algorithms represent individuals of the domain whereas a variable in our case represents a concept type in a locally correct tableau.

In order to avoid defining analogous expansion rules for different constraints with modal operators, we will use a notation that is similar to the unifying notation used in the properties of tableaux. To this end, α (and sometimes β) represents either a constraint of the form ϑ or a constraint of the form $x : C$. An expression of the form $[A]\alpha$ is either equal to some $[A]\vartheta$ or $x : [A]C$, and $\langle A\rangle\alpha$ to some $\langle A\rangle\vartheta$ or $x : \langle A\rangle C$. If $[A]\alpha = [A]\vartheta$ or $\langle A\rangle\alpha = \langle A\rangle\vartheta$, then $\alpha = \vartheta$; and if $[A]\alpha = x : [A]C$ or $\langle A\rangle\alpha = x : \langle A\rangle C$, then $\alpha = x : C$.

A variable x *occurs* in S if either one of $x : C$, $(x, y) : R$, or $(y, x) : R$ is in S. x is *fresh* for S if x does not occur in S and $x > y$ for all y occurring in S. If $S \in \mathbf{T}$, then the definition of *occurs* and *fresh* are also extended for $\mathbf{T}$. We assume that when a variable x is introduced to S, the constraint $x : \top$ is also added to S. If $(x, y) : R \in S$ for any R, then y is called a *successor* of x w.r.t. S. A variable y is called a *R-successor* of x w.r.t. S if $(x, y) : R \in S$.

A variable x is *blocked* by another variable y w.r.t. a constraint system S if $\{C \mid x : C \in S\} \subseteq \{D \mid y : D \in S\}$ and $y < x$. S (and therefore $\mathbf{T}$ if $S \in \mathbf{T}$) is said to contain a *clash* if for some variable x and some concept C, $\{x : C, x : \neg C\} \subseteq S$, or if for some formula ϑ, $\{\vartheta, \neg\vartheta\} \subseteq S$.

Let S be a constraint system for a $\mathbf{CL}_{\mathcal{ALC}}$ formula φ. The equivalence relation $\sim_S$ on the set of variables occurring in S is defined by taking $x \sim_S y$ iff $\{C \mid x : C \in S\} = \{D \mid y : D \in S\}$. The equivalence class generated by x is denoted by $[x]_S$. Finally, $\sim(S) = \{min([x]_S) : C \mid x : C \in S\} \cup \{\vartheta \mid \vartheta \in S\}$.

Let S be a constraint system for a $\mathbf{CL}_{\mathcal{ALC}}$ formula φ. $S' \subseteq S$ is called a *modal saturation* in S if S' is equal to

1. $\{[A_1]\alpha_1, \ldots, [A_n]\alpha_n\}$ such that $a \in A_i \cap A_j$ implies $i = j$,

2. $\{\langle A\rangle\alpha, [A_1]\alpha_1, \ldots, [A_n]\alpha_n\}$ such that $a \in A_i \cap A_j$ implies $i = j$ and $\bigcup_{i=1}^{n} A_i \subseteq A$, or,

3. $\{\langle A\rangle\alpha\}$.

A modal saturation S' in S is *maximal* if there is no other modal saturation S'' in S with $S'' \supset S'$. Let S be a modal saturation. Then $strip(S) = \{\alpha \mid [A]\alpha \text{ (or } \langle A\rangle\alpha) \in S\}$.

The tableau *expansion rules* are given in Figures 3 (local expansion rules) and 4 (the global expansion rule). A rule is *applicable* to a constraint system S if S satisfies the condition of the rule. A rule is applied to S if its action is executed due to the applicability of the rule to S.

Let φ be the $\mathbf{CL}_{\mathcal{ALC}}$ concept to be tested for satisfiability. The tableau algorithm starts with the completion set $\mathbf{T}_\varphi = \{S\}$, where $S = \{\varphi, x : \top\}$. $\mathbf{T}_\varphi$ is then expanded by repeatedly applying the rules in such a way that the global expansion rule is applied only when none of the local expansion rules is applicable to a constraint system. The expansion continues until the resulting completion set contains a clash or none of the rules is applicable to it. Such a completion set is called *complete*. If the expansion rules can be applied to $\mathbf{T}_\varphi$ in such a way that they yield a complete, clash-free constraint system, then the algorithm returns "φ is satisfiable", and "φ is unsatisfiable" otherwise.

The $\mathsf{R}_{\wedge}$ **rule**	
Condition:	$\vartheta_1 \wedge \vartheta_2 \in S$ and $\{\vartheta_1, \vartheta_2\} \not\subseteq S$.
Action:	Set $S = S \cup \{\vartheta_1, \vartheta_2\}$.
The $\mathsf{R}_{\vee}$ **rule**	
Condition:	$\vartheta_1 \vee \vartheta_2 \in S$ and $\{\vartheta_1, \vartheta_2\} \cap S = \emptyset$.
Action:	Set $S = S \cup \{\psi\}$ for some $\psi \in \{\vartheta_1, \vartheta_2\}$.
The $\mathsf{R}_{\sqcap}$ **rule**	
Condition:	$x : C_1 \sqcap C_2 \in S$ and $\{x : C_1, x : C_2\} \not\subseteq S$.
Action:	Set $S = S \cup \{x : C_1, x : C_2\}$.
The $\mathsf{R}_{\sqcup}$ **rule**	
Condition:	$x : C_1 \sqcup C_2 \in S$ and $\{x : C_1, x : C_2\} \cap S = \emptyset$.
Action:	Set $S = S \cup \{x : E\}$ for some $E \in \{C_1, C_2\}$.
The $\mathsf{R}_{\exists}$ **rule**	
Condition:	$x : \exists R.C \in S$, x is not blocked w.r.t. S, and x has no R-successor y w.r.t. S with $y : C \in S$.
Action:	Choose a fresh y for S and set $S = S \cup \{(x, y) : R, y : C\}$.
The $\mathsf{R}_{\forall}$ **rule**	
Condition:	$x : \forall R.C \in S$, there is a R-successor y of x w.r.t. S with $y : C \notin S$.
Action:	Set $S = S \cup \{y : C\}$.
The $\mathsf{R}_{=}$ **rule**	
Condition:	$C = \top \in S$ and $x : C \notin S$ for a variable x occurring in S.
Action:	Set $S = S \cup \{x : C\}$.
The $\mathsf{R}_{\neq}$ **rule**	
Condition:	$\neg(C = \top) \in S$ and there is no variable x such that $x : \dot{\neg} C \in S$.
Action:	Choose a fresh x for S and set $S = S \cup \{x : \dot{\neg} C\}$.
The $\mathsf{R}_{\langle \mathbb{A}\mathrm{gt} \rangle}$ **rule**	
Condition:	$\langle \mathbb{A}\mathrm{gt} \rangle \alpha \in S$ and $[\emptyset]\alpha \notin S$.
Action:	Set $S = S \cup \{[\emptyset]\alpha\}$.

Figure 3: Local expansion rules for $\mathbf{CL}_{\mathcal{ALC}}$.

The $\mathsf{R}_{[A\rangle}$ **rule**	
Condition:	$S_1, \ldots, S_n$ are all the maximal modal saturations in S.
Action:	Choose a fresh x for S, create sets $S_i' = \sim(strip(S_i) \cup \{x : \top\})$, where $1 \leq i \leq n$, and add them to $\mathbf{T}$.

Figure 4: The global expansion rule for $\mathbf{CL}_{\mathcal{ALC}}$.

4.2 Correctness and Termination

Theorem 10 (termination) *When started with the initial completion set* $\mathbf{T}_{\varphi}$*, the tableau algorithm terminates.*

Proof Contrary to [20] where the worst case complexity of the algorithm is established, here we give a rather general proof of termination.

Let $\mathbf{T}$ be the completion set for φ that is constructed by the algorithm from

$\mathbf{T}_\varphi$ and S_j an element of $\mathbf{T}$ with $1 \leq j \leq |\mathbf{T}|$. Denote by $L_j(x)$ the set of concepts $\{C \mid x : C \in S_j\}$. The *modal depth* $md(\psi)$ of ψ is the length of the longest chain of nested modal operators in ψ (both in subformulas and subconcepts). The modal depth $md(x : C)$ of a constraint $x : C$ is defined analogously. The modal depth $md(S_j)$ of a constraint system S_j is the maximal modal depth of constraints in S_j. The following properties can easily be derived from the definition of the algorithm:

1. The expansion rules never remove constraints from constraint systems or constraint systems from the completion set.
2. The number of subsets of $con^+(\varphi)$ is $2^{con^+(\varphi)}$, hence finite.
3. $|for+(\varphi)|$ is finite.

To prove that any sequence of rule applications is finite, it will be enough to show that there can only be finitely many constraint systems in $\mathbf{T}$ and finitely many variables in S_j. Let us first show that

(I) S_j can only have finitely many variables.

Consider all possible cases for variable introducing rules:

- $\mathsf{R}_\exists$: As there can only be a finite number of distinct $L_j(x)$ in S_j (by Property 2 above), a path of role successors will eventually get blocked. Hence the generation of a role path with infinite length is not possible.
- $\mathsf{R}_\neq$: As there can only be a finite number of constraints of the form $\neg(C = \top)$ in S_j (by Property 3 above), the number of $\mathsf{R}_\neq$ applications is limited in S_j.
- $\mathsf{R}_{[A\rangle}$: By the definition of this rule, the constraint system $S \subseteq S_j$ contains not more than $2^{con^+(\varphi)}$ distinct variables at the moment of its generation.

Now we show that the number of constraint systems in $\mathbf{T}$ should also be finite. From (I) and Property 2, we know that there are finitely many constraints of the form $x : [A_1]C$ and $y : \langle A_2\rangle D$ in S_j. Also, the number of modal formulas in S_j is finite due to Property 3. Hence, the maximal number of constraint systems generated by the global expansion rule from S_j is finite. Let S_l be such a constraint system. Clearly, $md(S_l) < md(S_j)$. Thus, it is not possible to have an infinite chain of constraint systems starting from S_j.

Theorem 11 (soundness) *If, when started with the initial completion set $\mathbf{T}_\varphi$ for a $\mathbf{CL}_{\mathcal{ALC}}$ formula φ, the expansion rules can be applied in such a way that they yield a complete and clash-free completion set, then there exists a locally correct tableau for φ.*

Proof Let $\mathbf{T}$ be the complete and clash-free completion set constructed by the tableau algorithm from $\mathbf{T}_\varphi$. A pentuple $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E} \rangle$ can be defined from $\mathbf{T}$ with:

1. $\Sigma = \{j \mid S_j \in \mathbf{T} \text{ for } 1 \leq j \leq |\mathbf{T}|\}$,
2. $\Lambda(j) = \{\psi \mid \psi \in S_j\}$,
3. $\mathbf{S}(j) = \{x \mid x \text{ occurs in } S_j \text{ and } x \text{ is not blocked w.r.t. } S_j\}$,
4. $\mathcal{L}_j(x) = \{C \mid x \in \mathbf{S}(j) \text{ and } x : C \in S_j\}$,
5. $\mathcal{E}_j(R)$ is equal to $\langle x, y \rangle \in \mathbf{S}(j) \times \mathbf{S}(j)$ such that
   (a) $(x, y) : R \in S_j$, or
   (b) $(x, z) : R \in S_j$ and y blocks z.

T satisfies properties (P1)-(P11) from Definition 14 because the expansion rules are not applicable to $\mathbf{T}$ in view of its completeness. To show that properties (P0), (P12)-(P14) hold, one must inductively construct an overrun o in T. As the base case, we assume for a random concept type $x' \in \mathbf{S}(j')$ that $x' \in o(j')$. Now two cases are possible.

1. Suppose $x \in o(j)$. If $[A]C$ or $\langle A \rangle C$ is in $\mathcal{L}_j(x)$, then there are states in Σ and concept types in those states which are introduced to $\mathbf{T}$ by the application of global expansion rules to x's constraints. Let k be such a state and $y_1, \ldots, y_n$ such concept types in $\mathbf{S}(k)$. We add y_i with $1 \leq i \leq n$ to $o(k)$ if y_i is not already in $o(k)$.
2. Suppose $x \in o(j)$, and $[A]C$ or $\langle A \rangle C$ is not in $\mathcal{L}_j(x)$. We know that in each constraint system there is a concept type introduced by an application of a global expansion rule. We choose a state k such that k is different from j and $o(k) = \emptyset$. Let R be the rule that added S_k to $\mathbf{T}$. We add the variable introduced by the application of R to $o(k)$.

As we apply the above cases inductively for each newly added variable to $o(k)$ then in the end o will return a non-empty set of concept types for each state, thus adhering to its definition. Since our choices in the beginning, namely x and $\mathbf{S}(j)$, are random, we can define an overrun starting with any concept type in any state. This means (P0) of Definition 14 is satisfied.

Theorem 12 (completeness) *If there exists a locally correct tableau for φ, when started with the initial completion set $\mathbf{T}_\varphi$, the expansion rules can be applied in such a way that the tableau algorithm yields a complete and clash-free completion set.*

Proof Let $T = \langle \Sigma, \Lambda, \mathbf{S}, \mathbf{O}, \mathcal{L}, \mathcal{E} \rangle$ be a locally correct tableau for φ. We use this tableau to guide the application of the non-deterministic rules to construct a complete and clash-free completion set for φ. Suppose that $\mathbf{T}$ is a completion set for φ. Define J as $\{j \mid S_j \in \mathbf{T} \text{ for } 1 \le j \le |\mathbf{T}|\}$ and say that $\mathbf{T}$ is T-*compatible* if the following holds:

1. there is a map σ from J to Σ such that if $\vartheta \in S_j$ then $\vartheta \in \Lambda(\sigma(j))$, for every $\vartheta \in \mathit{for}{+}(\varphi)$;
2. for each $j \in J$, there is a total function π_j from the set of variables in S_j to the set of concept types in $\mathbf{S}(\sigma(j))$ such that if $x : C \in S_j$ then $C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$, and if y is a R-successor of x w.r.t. S_j then $\langle \pi_j(x), \pi_j(y) \rangle \in \mathcal{E}_{\sigma(j)}(R)$.

Lemma 13 *If a completion set* $\mathbf{T}$ *for* φ *is* T*-compatible and* $\mathbf{T}'$ *is the result of an expansion rule* (R) *application to* $\mathbf{T}$*, then* $\mathbf{T}'$ *is* T*-compatible as well.*

Proof Proof of Lemma 13 Let $\mathbf{T}$ be a T-compatible completion set, S_j an element in $\mathbf{T}$, and σ and π_j the functions supplied by the definition of T-compatibility. Consider all possible cases for R.

- $\mathsf{R}_\wedge$: If $\vartheta_1 \wedge \vartheta_2 \in S_j$, then $\vartheta_1 \wedge \vartheta_2 \in \Lambda(\sigma(j))$. This implies $\vartheta_1, \vartheta_2 \in \Lambda(\sigma(j))$ due to (P9) from Definition 14. The application of $\mathsf{R}_\wedge$ to S_j adds ϑ_1 and ϑ_2 to S_j. Hence the rule guarantees that $\mathbf{T}'$ is T-compatible.
- $\mathsf{R}_\vee$: If $\vartheta_1 \vee \vartheta_2 \in S_j$, then $\vartheta_1 \vee \vartheta_2 \in \Lambda(\sigma(j))$. Since T is a locally correct tableau, (P10) from Definition 14 implies ϑ_1 or ϑ_2 is in $\Lambda(\sigma(j))$. If $\vartheta_1 \in \Lambda(\sigma(j)$ then apply the rule so that ϑ_1 is added to S_j, else apply the rule so that ϑ_2 is added to S_j. Hence $\mathsf{R}_\vee$ can be applied in such a way that it preserves T-compatibility.
- $\mathsf{R}_\sqcap$: If $x : C_1 \sqcap C_2 \in S_j$, then $C_1 \sqcap C_2 \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$. This implies $C_1, C_2 \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$ due to (P2) from Definition 14. The application of $\mathsf{R}_\sqcap$ to S_j adds $x : C_1$ and $x : C_2$ to S_j. Hence the rule guarantees that $\mathbf{T}'$ is T-compatible.
- $\mathsf{R}_\sqcup$: In this case, the rule is applied to $x : C_1 \sqcup C_2 \in S_j$. Since T is a locally correct tableau, (P3) from Definition 14 implies C_1 or C_2 is in $\mathcal{L}_{\sigma(j)}(\pi_j(x))$. If $C_1 \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$ then apply the rule so that $x : C_1$ is added to S_j, else apply the rule so that $x : C_2$ is added to S_j. Hence $\mathsf{R}_\sqcup$ can be applied in such a way that it preserves T-compatibility.
- $\mathsf{R}_\exists$: If $x : \exists R.C \in S_j$, then $\exists R.C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$. Since T is a locally correct tableau, (P5) of Definition 14 implies that there is an element $t \in \mathbf{S}(\sigma(j))$ such that $\langle \pi_j(x), t \rangle \in \mathcal{E}_{\sigma(j)}(R)$ and $C \in \mathcal{L}_{\sigma(j)}(t)$. The application of $\mathsf{R}_\exists$ introduces a new variable y with $(x, y) : R \in S_j$ and $y : C \in S_j$. Hence we set $\pi'_j = \pi_j[y \mapsto t]$. The function π'_j is then as required for the resulting completion set $\mathbf{T}'$.

- $\mathsf{R}_\forall$: If $x : \forall R.C \in S_j$ and y is a R-successor of x w.r.t. S_j, then $\forall R.C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$ and $\langle \pi_j(x), \pi_j(y) \rangle \in \mathcal{E}_{\sigma(j)}(R)$ due to T-compatibility. Since T is a locally correct tableau, (P4) of Definition 14 implies $C \in \mathcal{L}_{\sigma(j)}(\pi_j(y))$. The rule adds $y : C$ to S_j and thus guarantees that $\mathbf{T}'$ is T-compatible.
- $\mathsf{R}_=$: If $C = \top \in S_j$, then $C = \top \in \Lambda(\sigma(j))$. Since T is a locally correct tableau, (P6) of Definition 14 implies for every x occurring in S_j that $C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$. The rule adds $x : C$ to S_j for every such x and thus guarantees that $\mathbf{T}'$ is T-compatible.
- $\mathsf{R}_{\neq}$: If $\neg(C = \top) \in S_j$ then $\neg(C = \top) \in \Lambda(\sigma(j))$. Since T is a locally correct tableau, (P7) of Definition 14 implies that there is a $s \in \mathbf{S}(\sigma(j))$ with $\dot\neg C \in \mathcal{L}_{\sigma(j)}(s)$. The application of $\mathsf{R}_{\neq}$ introduces a new variable x with $x : \dot\neg C \in S_j$. Hence we set $\pi'_j = \pi_j\,[x \mapsto s]$. The function π'_j is then as required for the resulting completion set $\mathbf{T}'$.
- $\mathsf{R}_{\langle \mathbb{A}\mathrm{gt} \rangle}$: We only prove the case for when $\langle \mathbb{A}\mathrm{gt} \rangle \alpha = x : \langle \mathbb{A}\mathrm{gt} \rangle C$, because the formula case can be treated analogously. $x : \langle \mathbb{A}\mathrm{gt} \rangle C \in S_j$ implies $\langle \mathbb{A}\mathrm{gt} \rangle C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$. Since T is a locally correct tableau, by (P11) of Definition 14 we have $[\emptyset]C \in \mathcal{L}_{\sigma(j)}(\pi_j(x))$. The rule adds $x : [\emptyset]C$ to S_j and thus guarantees that $\mathbf{T}'$ is T-compatible.
- $\mathsf{R}_{[A]}$: If $\mathsf{R}_{[A]}$ is applicable to $\Psi = \{[A_1]\alpha_1, \ldots, [A_n]\alpha_n\} \subseteq S_j$, then by definition $[A_1], \ldots, [A_n]$ are disjoint coalitions. Let $\Pi_j(\Psi) = \{[A]\vartheta \mid [A]\vartheta \in \Psi\} \cup \{\pi_j(x) : [A]C \mid x : [A]C \in \Psi\}$. Clearly, $\Pi_j(\Psi) \subseteq \Phi_{\sigma(j)}$. Since T is a locally correct tableau, by (P11) in Definition 14 there is some $v \in \Sigma$ such that $v \in \mathsf{Ins}(\Pi_j(\Psi))$. The application of $\mathsf{R}_{[A]}$ introduces a new constraint system S_l. Hence we set $\sigma' = \sigma[l \mapsto v]$. We distinguish two cases for a $[A_i]\alpha_i \in \Psi$.

  1. If $[A_i]\alpha_i = [A_i]\vartheta$ then $\vartheta \in S_l$ (due to $\mathsf{R}_{[A]}$'s definition) which means σ' is as required for T-compatibility i.e. $\vartheta \in \Lambda(\sigma'(l))$.
  2. If $[A_i]\alpha_i = x : [A_i]C$ then there should be a concept type $t \in \mathbf{S}(v)$ such that $\{C \mid x : [A]C \in \Psi\} \subseteq \mathcal{L}_v(t)$, $\pi_j(x) \in o(w)$, and $t \in o(v)$ (due to (P11) of Definition 14). $\mathsf{R}_{[A]}$ guarantees that there is a variable y such that $\{C \mid x : [A]C \in \Psi\} = \{C \mid y : C \in S_l\}$. Set $\pi_l(y) = t$.

  Finally, $\mathsf{R}_{[A]}$ makes sure that there is at least one variable, say z, occuring S_l. Hence, we set $\pi_l(z) = s'$ for a concept type $s' \in \mathbf{S}(v)$. Such a s' exists because by Definition 12, $\mathbf{S}(v)$ is non-empty. The functions σ' and π_l are then as required for the resulting completion set $\mathbf{T}'$.
- The arguments for $\mathsf{R}_{[A\rangle}$ and $\mathsf{R}_{\langle A\rangle}$ are analogous to $\mathsf{R}_{[A]}$.

Now we show that the completeness of the tableau algorithm follows from the lemma above. Let S_1 be the (initial) constraint system in $\mathbf{T}_\varphi$, and x the variable in S_1. Set $\sigma(1) = w_\varphi$ and $\pi_1(x) = s$ for a $s \in \mathbf{S}(w_\varphi)$ (such w_φ and s

exist since T is a locally correct tableau for φ). It is easy to see that these functions are as needed for $\mathbf{T}_\varphi$'s T-compatibility. We know by the claim above that whenever a rule is applicable to $\mathbf{T}_\varphi$, it can be applied in a way that it maintains T-compatibility. Also, from Theorem 10, any sequence of rule applications must terminate. Thus, we have eventually a completion set $\mathbf{T}$ that is T-compatible. This completion set must be clash-free.

Suppose otherwise. Let S_j be a constraint system in $\mathbf{T}$ such that $\{x : C, x : \neg C\} \subseteq S_j$. Then we have $\{C, \neg C\} \subseteq \mathcal{L}_{\sigma(j)}(\pi_j(x))$ which violates Property (P1) in Definition 14. A similar argument can be made for a clash of the form $\{\vartheta, \neg\vartheta\} \subseteq S_j$.

5 Conclusions

In this paper, we introduce the coalitional description logic $\mathbf{CL}_{\mathcal{ALC}}$ and present a tableau decision procedure for its constant domain variant. To our best knowledge, this is the first formal study of a logic that combines DL perspective with strategic modalities. Therefore, the paper can be seen as an initiative to integrate the game-theoretic dimension with description logics. Alternatively, one can see our proposal as an attempt to extend the agenda of modal logics of strategies to reasoning about individuals and their classes without losing decidability.

We believe that our work can be useful for the semantic web community. One of the most interesting problems in this area is to discover (in an automated way) a sequence of service executions that will satisfy a user's goals. Current approaches are mainly based on standard DL subsumption testing of the desired input and output classes with the advertised input and output classes of web services [21]. On the other hand, agent logics provide well studied semantics of time, action, and strategy execution, that can be used in reasoning about web services.

Our results are also interesting algorithmically. An important property regarding the optimization of our decision procedure for $\mathbf{CL}_{\mathcal{ALC}}$ is that once the application of expansion rules to a constraint system has been exhausted, then the algorithm can simply discard that constraint system from the memory. This is a unique feature for a constant domain modal DL, and the major difference from the decision procedure for $\mathbf{K}_{\mathcal{ALC}}$ [20], i.e., the normal modal logic extension of $\mathcal{ALC}$. Moreover, our algorithm does not need marked variables and the non-deterministic rules which make use of them in [20]. This difference is due to the lack of the notion of accessibility between worlds (more precisely, the non-normal semantics of the necessitation operator). It would be an interesting line of work to investigate these logics complexity-wise.

The next step at this point would be to extend the tableau algorithm for

CL$_{\mathcal{ALC}}$ to **ATL**$_{\mathcal{ALC}}$. **ATL** [1, 2] expands the language of **CTL** by allowing a path quantifier for each coalition of agents, and **CL** can be seen as a fragment of **ATL** [12]. The satisfiability problem for **ATL** is known to be EXPTIME-complete [29]. The decidability result of a product of **PDL** (which is also EXPTIME-complete) with $\mathcal{ALC}$ [30] gives us a reason to believe that **ATL**$_{\mathcal{ALC}}$ is decidable.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press, 1997.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[3] F. Baader et al., editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[4] F. Baader, R. Küsters, and F. Wolter. Extensions to description logics. In Baader et al. [3], pages 219–261.

[5] F. Baader and A. Laux. Terminological logics with modal operators. In C. Mellish, editor, *14th International Joint Conference on Artificial Intelligence*, pages 808–814, Montreal, Canada, 1995. Morgan Kaufman.

[6] F. Baader and W. Nutt. Basic description logics. In Baader et al. [3], pages 43–95.

[7] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[8] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3):213–261, 1990.

[9] O. Dikenelli, R. C. Erdur, G. Kardas, Ö. Gümüs, I. Seylan, Ö. Gürcan, A. M. Tiryaki, and E. E. Ekinci. Developing multi agent systems on semantic web environment using seagent platform. In O. Dikenelli, M. P. Gleizes, and A. Ricci, editors, *ESAW*, volume 3963 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2005.

[10] R. C. Erdur and İ. Seylan. The design of a semantic web compatible content language for agent communication. *Expert Systems*, 25(3):268–294, July 2008.

[11] FIPA. *FIPA Communicative Act Library Specification*. FIPA, 2002.

[12] V. Goranko. Coalition games and alternating temporal logics. In *TARK '01: Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge*, pages 259–272, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[13] H. H. Hansen. Tableau games for coalition logic and alternating-time temporal logic. Master's thesis, Universiteit van Amsterdam, 2004.

[14] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[15] I. Horrocks, U. Sattler, and S. Tobies. A description logic with transitive and converse roles, role hierarchies and qualifying number restrictions. LTCS-Report LTCS-99-08, LuFG Theoretical Computer Science, RWTH Aachen, 1999. Revised version. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[16] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *LPAR '99*, pages 161–180, London, UK, 1999. Springer-Verlag.

[17] A. Laux. Beliefs in multi-agent worlds: a terminological logics approach. In *11th European Conference on Artificial Intelligence*, pages 299–303, Amsterdam, The Netherlands, 1994.

[18] A. Laux. Integrating a modal logic of knowledge into terminological logics. In H. Geffner, editor, *Proceedings of the Fourth Iberoamerican Congress on Artificial Intelligence*, pages 76–92, Caracas, Venezuela, 1994.

[19] C. Lutz, H. Sturm, F. Wolter, and M. Zakharyaschev. Tableaux for temporal description logic with constant domains. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR*, volume 2083 of *Lecture Notes in Computer Science*, pages 121–136. Springer, 2001.

[20] C. Lutz, H. Sturm, F. Wolter, and M. Zakharyaschev. A tableau decision algorithm for modalized $\mathcal{ALC}$ with constant domains. *Studia Logica*, 72(2):199–232, 2002.

[21] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic matching of web services capabilities. In I. Horrocks and J. A. Hendler, editors, *ISWC*, volume 2342 of *LNCS*, pages 333–347. Springer, 2002.

[22] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, October 2001.

[23] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, February 2002.

[24] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991.

[25] M. D. Sadek. A study in the logic of intention. In *KR*, pages 462–473, 1992.

[26] L. Sauro, J. Gerbrandy, W. van der Hoek, and M. Wooldridge. Reasoning about action and cooperation. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *AAMAS*, pages 185–192. ACM, 2006.

[27] İ. Seylan and R. C. Erdur. A tableau decision procedure for $\mathcal{ALC}$ with monotonic modal operators and constant domains. In C. Areces and S. Demri, editors, *M4M*, 2007.

[28] H. Sturm and F. Wolter. A tableau calculus for temporal description logic: the expanding domain case. *J. Log. Comput.*, 12(5):809–838, 2002.

[29] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *LICS '03: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*, page 208, Washington, DC, USA, 2003. IEEE Computer Society.

[30] F. Wolter and M. Zakharyaschev. Dynamic description logics. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *AIML*, pages 431–446. CSLI Publications, 1998.

[31] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In S. S. A.G. Cohn, L. Schubert, editor, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 512–523, Montreal, Canada, 1998. Morgan Kaufman.

[32] F. Wolter and M. Zakharyaschev. Modal description logics: modalizing roles. *Fundam. Inf.*, 39(4):411–438, 1999.

[33] F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In T. Dean, editor, *IJCAI*, pages 104–109. Morgan Kaufmann, 1999.

[34] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems*, pages 379–402. Studies Press/Wiley, 1999.

[35] M. Wooldridge. Semantic issues in the verification of agent communication languages. *Autonomous Agents and Multi-Agent Systems*, 3(1):9–31, 2000.

[36] M. Wooldridge, T. Agotnes, P. E. Dunne, and W. van der Hoek. Logic for automated mechanism design – a progress report. In *AAAI-07*, Vancouver, Canada, July 2007.